



SAPIENZA UNIVERSITÀ DI ROMA
DOTTORATO DI RICERCA IN FISICA
SCUOLA DI DOTTORATO "VITO VOLTERRA"

Theoretical analysis of optimization problems

THESIS SUBMITTED TO OBTAIN THE DEGREE OF
"Dottore di Ricerca" - Doctor Philosophiæ
PHD IN PHYSICS - XX CYCLE - OCTOBER 2007

BY

Fabrizio Altarelli

Program Coordinator

Prof. Enzo Marinari

Thesis Advisors

Prof. Giorgio Parisi

Dr. Nicolas Sourlas

Dr. Rémi Monasson

Alla dottoressa Federici

Contents

Introduction	vii
I Statistical mechanics of optimization problems	1
1 Statistical mechanics of disordered systems	3
1.1 Statistical mechanics and phase transitions	3
1.1.1 The Gibbs distribution	3
1.1.2 Phase transitions and ergodicity breaking	5
1.2 Disordered systems and spin glasses	6
1.2.1 Origins of disorder	6
1.2.2 Spin glass models	7
1.2.3 Mean field theory and diluted models	8
1.2.4 Frustration, local degeneracies, complexity	8
1.2.5 The order parameter of disordered systems	9
1.3 Phenomenology of disordered systems	10
1.3.1 Spin glass susceptibilities	10
1.3.2 Divergence of relaxation times	12
1.3.3 Ageing	13
1.4 The replica method	13
1.4.1 The replica trick	13
1.4.2 Solution of the p -spin spherical model	14
1.4.3 Replica formalism for diluted models	17
2 Optimization problems and algorithms	21
2.1 Some examples of combinatorial optimization problems	21
2.2 Boolean satisfiability: k -SAT and k -XORSAT	23
2.2.1 Introduction to k -SAT	23
2.2.2 Introduction to k -XORSAT	26
2.3 Computational complexity	26
2.3.1 Algorithms and computational resources	27
2.3.2 Computation models and complexity classes	27
2.3.3 Reductions, hardness and completeness	30
2.3.4 Other measures of complexity	31
2.3.5 Connections to the work presented in Part II	33
2.4 Search algorithms	33

2.4.1	Random-walk algorithms	34
2.4.2	DPLL algorithms	36
3	Phase transitions in random optimization problems	43
3.1	Evidence of phase transitions from numerical experiments	43
3.2	Rigorous derivation of the phase diagram of k -XORSAT	44
3.2.1	Bounds from first and second moments	45
3.2.2	Leaf removal procedure	47
3.2.3	Phase diagram of k -XORSAT	51
3.3	Heuristic results on the phase diagram of k -SAT	52
II	Some properties of random k-SAT and random k-XORSAT	55
4	Study of poissonian heuristics for DPLL in k-XORSAT	57
4.1	Leaf-removal for mixed formulæ	58
4.1.1	Leaf-removal differential equations	58
4.1.2	Solution for $c_j(t)$	59
4.1.3	Solution for $n_\ell(t)$	60
4.2	Characterization of the phases in terms of a potential	61
4.2.1	Definition and properties of the potential $V(b)$	61
4.2.2	Phase diagram for mixed k -XORSAT formulæ	63
4.3	Trajectories generated by poissonian heuristics	64
4.3.1	Poissonian heuristics for DPLL	65
4.3.2	General properties of poissonian heuristics	67
4.3.3	Analysis of UC and GUC	71
4.4	Bounds on the values of α for which poissonian heuristics can succeed	73
4.5	Optimality of GUC for large k	74
4.6	Conclusions and perspectives	76
5	Characterization of the solutions of k-SAT at large α	79
5.1	Problem definition and previously established results	80
5.1.1	Definition of the random ensembles	80
5.1.2	Hardness of approximation results	80
5.1.3	Performance of WARNING PROPAGATION on the planted distribution	83
5.1.4	Discussion of the known results and problem definition	85
5.2	Free energy of the uniform distribution of satisfiable formulæ	85
5.2.1	Replicated partition function of k -SAT	86
5.2.2	Free energy and replica symmetric ansatz	87
5.2.3	Selection of satisfiable formulæ by means of a “chemical potential”	88
5.2.4	Saddle point equations	92
5.2.5	Distribution of fields	93
5.2.6	Ground state energy	94
5.3	Cavity formalism for the fields distribution	97
5.4	Comparison of \mathcal{P}_{Sat} and $\mathcal{P}_{\text{Plant}}$ at large α	100
5.4.1	Distribution of fields	100
5.4.2	Correlation between field and number of occurrences	101

5.4.3	Finite energy results	102
5.4.4	Algorithmic implications	102
5.5	Stability of the RS free energy	103
5.5.1	Solutions with non-integer fields	103
5.5.2	Eigenvalues of the stability matrix	104
5.5.3	Uniqueness of the solution	107
5.6	Discussion of the results and conclusion	112
Acknowledgements		114
List of notations		117

Introduction

The mean field theory of disordered systems is a well established topic in statistical mechanics, developed in the past thirty years with remarkable success (see [1] for the classical reference). Originally, interest in this field was motivated by the experimental discovery of *spin glasses*, metallic compounds formed by diluting a ferromagnetic metal in a diamagnetic host, and which exhibit peculiar magnetic properties: on one hand, the low dilution ensures that the locations of the ferromagnetic atoms (and therefore their interactions) are random, so that their structure presents no order; on the other hand, evidence is found at low temperature for a transition to a phase in which the local magnetization is frozen (in a direction variable from point to point) and which therefore displays some properties characteristic of the presence of order. The phenomenology of spin glasses is indeed very rich, and the development of theoretical models able to explain and reproduce it in full has been a major challenge (and achievement) of statistical mechanics in the past thirty years, requiring the introduction of innovative concepts and techniques.

During the same time span, a large number of interesting results have been obtained in the understanding of *combinatorial optimization problems*, and in the development of *computational complexity theory* (see [2] for an introduction). Combinatorial optimization problems (that is to say, problems in which the *optimal* configuration in a large and discrete set of candidates has to be found) are of the greatest interest for practical applications, and turn out to be general enough to deserve considerable attention from the theoretical point of view as well. Moreover, they are the cornerstone of complexity theory, the purpose of which is to characterize the intrinsic “hardness” of solving problems, and also the efficiency of algorithms used to solve them.

Very early, it was recognized that these two fields, apparently far from each other, and studied by different communities of researchers, actually have very much in common. It was soon realized that random distributions of some well known and very important (both theoretically and in view of applications) combinatorial optimization problems were formally equivalent to diluted spin glass models, and could be treated with such powerful tools as the replica method and (somewhat later) the cavity approach. This has led, in the past decade, to a very fecund transfer of problems and ideas across the two fields, leading to significant advances in our understanding of both.

A first area of interest is the characterization of the different phases of models that are relevant from both the optimization and statistical mechanics points of view. These models consist of a collection of N Ising spins that interact with k -body couplings ($k = 2, 3, \dots$) with random strenghts (the exact form of the interactions defines each model). The number of interactions to which individual spins participate, also called *connectivity*, plays the role of the control parameter, analogous to the pressure in thermodynamic systems. As the connectivity is varied, the free energy “landscape” undergoes a series of dramatic structural changes, that correspond to the onset different “macroscopic” properties of the system, such as the presence of an exponential (in N) number of local minima in the landscape, or the value of the energy density of the global ground state being of order 1 rather than order N .

Another area of interest is the analysis of the algorithms that can be used in order to solve the optimization problem represented by each model, or in equivalent but more physical terms, to find its ground state configurations. There is a huge variety in these algorithms: some of them define a *dynamical* process which modifies the configuration, in a manner similar to the well known Metropolis algorithm; some others perform a *sequential assignment* of the values of spins trying to minimize the number of positive contributions to the hamiltonian; others still do not act on the spins themselves, but rather on some *effective variables*, such as the magnetic field conjugated to each spin. Accordingly, a full “taxonomy” of algorithms can be constructed, and the average behaviour of whole classes of algorithms, with similar structure but different attributes, can be characterized, allowing both to identify those algorithms that are of interest from the point of view of actual applications, and also to reach a better understanding in the properties of the models themselves.

Even though the list of the topics that have been studied in this field, and which are of interest for current research, includes many more, I shall limit the discussion to the previous ones: in this thesis, I have worked on problems that stem from these two lines of research. In the first Part, I shall therefore introduce the models I have studied and give an overview of the most relevant known results. Chapter 1 is an introduction to the physics of disordered systems: the main concepts of the statistical mechanics of spin glasses are introduced, with a discussion of the main phenomenological features that characterize them, and of the replica method, which allows to study them analytically. In Chapter 2, I shall introduce combinatorial optimization problems, and some important results from the theory of computational complexity that are relevant to my work; in particular, the two boolean satisfiability problems that I have been interested in, called k -SAT and k -XORSAT, are defined, and their properties are discussed. Finally, in Chapter 3 I shall review the results obtained by applying the methods and concepts developed for spin glasses to these two problems, and what their interpretation as spin glasses can teach us about the physics of these and similar systems.

In the second Part of the thesis, I shall present some of the original results that I have obtained in collaboration with Rémi Monasson, Giorgio Parisi and Francesco Zamponi.

The first problem we have studied is motivated by a well known (but not as well understood) empirical observation: a large variety of systems present a phase in which the ground states form *clusters* and the spins are *frozen*; in this phase, no *local* search algorithm is capable to find the ground states in an *efficient* manner. In this context (and very loosely speaking), clusters are sets of configurations which all have the ground state energy and which are connected, while different clusters are well separated (two configurations are considered *adjacent* if they differ by a number of spins which is of order 1 in N , *connected* if one can be reached from the other with a series of adjacent steps, and *separated* if this is not possible); a frozen spin is a spin that takes the same value in all the configurations of a cluster; a local search algorithm is an algorithm that only uses information about the values of a number of variables which is of order 1 in N ; and efficient means that the time (or number of elementary computations) required to find a ground state configuration with this algorithm grows faster than any polynomial in N .

The simplest model presenting such a “clustered-frozen” phase is k -XORSAT, which I mentioned before and I shall discuss in Chapter 2; on the other hand, one of the most studied (and useful in practical applications) local algorithms is DPLL, which works by assigning variables in sequence according to some simple rule called heuristics, and which I shall also discuss in Chapter 2. In order to gain a better understanding of the failure of local search algorithms in the clustered-frozen phase, we have studied a fairly general class of DPLL heuristics for k -XORSAT, obtaining some results that I shall present in Chapter 4. Most notably, we have obtained the first proof (to the best of my knowledge)

that any heuristic in this class fails to find a ground state in polynomial time in N with probability 1 as N goes to infinity. Moreover, we have obtained an argument that supports the claim that in the large k limit, one of the heuristics belonging to the class we have studied (and which was previously introduced and called GUC) is capable of finding ground states efficiently with probability 1 up to the onset of the clustered-frozen phase, while all the other heuristics previously studied were known to fail well before this phase transition.

The second problem we have considered concerns the most studied and celebrated combinatorial optimization problem: k -SAT. There are many reasons motivating the interest for k -SAT, notably that it is the first problem for which NP-completeness (which is the key concept in computational complexity theory) was proven, and that it is so general that a huge number of other problems (many of which are relevant in view of applications) can be expressed as particular instances of k -SAT. As a result of these extended studies, a very rich phase structure has emerged, with a multitude of transitions determined by temperature and connectivity. The aim of our work was to study the phase which is obtained at zero temperature when the connectivity goes to infinity. Apart from the intrinsic interest of studying one of the phases of the system, this problem is very interesting due to some recent results in computational complexity theory that establish a link between the *average* case complexity of k -SAT at large connectivity and the *worst* case complexity of several other problems. No relation between these two measures of complexity was previously known, and the complexity class of the problems considered depends on the properties of k -SAT at large connectivity.

The main result we have obtained is that this phase of the system is characterized by the presence of a single cluster of ground states in which the fraction of spins that are not frozen goes exponentially to 0 as the connectivity is increased, and that the field conjugated to frozen spins is of the same order of the connectivity. I shall present these results in Chapter 5, together with a discussion of their interest and consequences for computational complexity theory.

Moreover, during the past year I have engaged in the study of yet another algorithm for boolean satisfiability problems, going under the name of WalkSAT. This work, which consists in a numerical characterization of the average behavior of the algorithm, and in elucidating the properties of k -SAT that this behavior imply, is still in progress, and will constitute the object of a future publication.

Part I

Statistical mechanics of optimization problems

Chapter 1

Statistical mechanics of disordered systems

1.1 Statistical mechanics and phase transitions

In this section I shall introduce some notation and briefly review some fundamental concepts of statistical mechanics, illustrating them with the example of the Ising ferromagnet.

1.1.1 The Gibbs distribution

A general system studied in statistical mechanics will have a large number N of degrees of freedom $\{x_i \in \mathbb{X} \mid i = 1, \dots, N\}$. A configuration $\mathcal{C} \in \mathbb{X}^N$ of the system is determined by specifying the value taken by each x_i . The hamiltonian of the system will be an extensive function of the configuration, $H(\mathcal{C})$.

The statistical properties of the system are determined by the probability distribution of the configurations. If a system can exchange energy with its surrounding at temperature¹ $T \equiv 1/\beta$, this probability is given by the Gibbs distribution:

$$\mathbb{P}[\mathcal{C}] = \frac{1}{Z(\beta)} e^{-\beta H(\mathcal{C})} \quad (1.1)$$

where the partition function $Z(\beta)$ is a normalization. In fact, it is much more than a normalization, since all the equilibrium properties of the system can be computed from it. For example, the average moments of the energy are given by its derivatives:

$$E(\beta) \equiv \mathbb{E}[H(\mathcal{C})] = -\frac{\partial}{\partial \beta} \log Z(\beta), \quad (1.2)$$

$$\mathbb{E}[H(\mathcal{C})^2] - \mathbb{E}[H(\mathcal{C})]^2 = \frac{\partial^2}{\partial \beta^2} \log Z(\beta), \dots \quad (1.3)$$

The entropy and the free energy can be introduced in two equivalent ways. The “microcanonical” entropy is the logarithm of the number of configurations with energy E :

$$S_m(E) = \log |\{\mathcal{C} \in \mathbb{X}^N : H(\mathcal{C}) = E\}|. \quad (1.4)$$

¹I shall always use “natural” units, in which the Boltzmann constant is equal to 1.

We can expect it to be an extensive quantity and define the entropy density $s_m(e) = S_m(Ne)/N$. Since the Gibbs measure depends on the configurations only through the energy, we can greatly simplify the description of the system by considering the probability to find it in *any* configuration of energy E :

$$\mathbb{P}[E] = \sum_{\{\mathcal{C} \in \mathbb{X}^N | H(\mathcal{C})=E\}} \frac{1}{Z(\beta)} e^{-\beta H(\mathcal{C})} = \frac{1}{Z(\beta)} e^{-\beta E + S_m(E)} \equiv \frac{1}{Z(\beta)} e^{-\beta F_m(E)} \quad (1.5)$$

where we have introduced the free energy $F_m(E) \equiv N f_m(E/N) \equiv E - S_m(E)/\beta$.

On the other hand, the “canonical” entropy is defined in terms of the Gibbs distribution as

$$S_c(\beta) \equiv -\mathbb{E}[\log \mathbb{P}[\mathcal{C}]] = -\sum_{\mathcal{C}} \mathbb{P}[\mathcal{C}] \log \mathbb{P}[\mathcal{C}], \quad (1.6)$$

(notice that $\mathbb{P}[\mathcal{C}]$ depends on β) while the free energy is defined

$$F_c(\beta) \equiv -\frac{1}{\beta} \log Z(\beta). \quad (1.7)$$

Notice that these definitions imply that

$$S_c(\beta) = -\mathbb{E} \left[\log \frac{e^{-\beta H(\mathcal{C})}}{Z(\beta)} \right] = \beta E(\beta) - \beta F_c(\beta) \quad (1.8)$$

which formally corresponds to the similar microcanonical relation.

The relationship between the microcanonical and canonical approaches becomes evident in the thermodynamic limit $N \rightarrow \infty$. In this limit, we can compute the canonical free energy with the Laplace method:

$$f_c(\beta) = -\lim_{N \rightarrow \infty} \frac{1}{N} \frac{1}{\beta} \log Z(\beta) \quad (1.9)$$

$$= -\lim_{N \rightarrow \infty} \frac{1}{N} \frac{1}{\beta} \log \int de e^{-N\beta f_m(e)} \quad (1.10)$$

$$= -\lim_{N \rightarrow \infty} \frac{1}{N} \frac{1}{\beta} \log e^{-N\beta f_m(\bar{e})} \quad (1.11)$$

$$= f_m(\bar{e}) \quad (1.12)$$

where \bar{e} is the value that maximizes the exponent, i.e. $f'_m(\bar{e}) = 0 \Leftrightarrow s'_m(\bar{e}) = \beta$. But in the thermodynamic limit the energy is concentrated, so that

$$e(\beta) = \lim_{N \rightarrow \infty} \int de e \frac{e^{-N\beta e + N s_c(e)}}{Z(\beta)} = \bar{e} e^{-\beta \bar{e} + s_m(\bar{e}) - \beta f_m(\bar{e})} = \bar{e} \quad (1.13)$$

from (1.7) and (1.12). Therefore $f_c(\beta) = f_m(e(\beta))$ and $s_c(\beta) = s_m(e(\beta))$.

The physical interpretation of the free energy becomes clear by observing that (1.5) can be rewritten as

$$\mathbb{P}[e] = \frac{1}{Z(\beta)} e^{-N\beta f_m(e)} = e^{-N\beta [f_m(e) - f_m(e(\beta))]}, \quad (1.14)$$

i.e. the probability that e takes a value which is different from the expected value is exponentially small in N and the corresponding large deviations function is the free energy itself.

Also notice that if the energy of a configuration \mathcal{C} only depends on some extensive observable O , i.e. $H(\mathcal{C}) = \mathcal{E}(O(\mathcal{C}))$ where \mathcal{E} is some function, then the expected value and the distribution of the large deviations of O can be expressed in a similar way in terms of the free energy, by writing it as a function of $o \equiv O/N$.

1.1.2 Phase transitions and ergodicity breaking

Let us now discuss a specific example: the infinite range Ising ferromagnet. The degrees of freedom are N Ising spins $\sigma_i \in \{-1, 1\}$. We consider that each spin interacts with all the others and with a homogeneous external field h^{ext} :

$$H(\mathcal{C}) = - \sum_{i < j}^{1, N} J_N \sigma_i \sigma_j - \sum_i h^{\text{ext}} \sigma_i. \quad (1.15)$$

In order for the energy to be extensive, J_N must scale with the number of spins as J/N , and we set the energy units so that the factor J is 1.

It is easy to solve this model with the trick discussed in the last paragraph of the previous section: the energy (1.15) depends on the configuration only through the total magnetization $M(\mathcal{C}) = \sum_i \sigma_i$, which is an extensive quantity. In terms of densities

$$e(m) = -\frac{1}{2}m^2 - h^{\text{ext}}m. \quad (1.16)$$

The number of configurations with magnetization M is just $\binom{N}{N_+}$ where $N_+ = (N + M)/2$ is the number of up spins, so that the (microcanonical) entropy is obtained by Stirling's approximation:

$$s(m) = -\frac{1+m}{2} \log \frac{1+m}{2} - \frac{1-m}{2} \log \frac{1-m}{2}. \quad (1.17)$$

The equilibrium magnetization \bar{m} is obtained introducing $f(m) = e(m) - s(m)/\beta$ from the condition

$$f'(\bar{m}) = 0 \Leftrightarrow -\bar{m} - h^{\text{ext}} - \frac{1}{2\beta} [\log(1+\bar{m}) - \log(1-\bar{m})] = 0, \quad (1.18)$$

from which the self-consistent equation

$$\bar{m} = \tanh[\beta(\bar{m} + h^{\text{ext}})] \quad (1.19)$$

is found.

We see that for $\beta > 1$ this equation admits a solution with $\bar{m} \neq 0$ even if $h^{\text{ext}} = 0$, i.e. there is a spontaneous magnetization, while for $\beta < 1$ this is not the case. This is one of the simplest examples of phase transition, in which the magnetization has the role of the *order parameter* characterizing the phases. Notice that the existence of a spontaneous magnetization is a very striking phenomenon: in the absence of an external field, the energy is an even function of the magnetization, and the Gibbs weight of the configurations with magnetization m is the same as that corresponding to magnetization $-m$, so that the expected value of the magnetization is 0 *at all temperatures*.

The solution of this apparent contradiction can be understood by a more careful consideration the free energy of the problem. In the absence of field, $f(m)$ is an even function of m . It can be easily seen that the sign of $f''(0)$ is the same as that of $1 - \beta$: at high temperature $m = 0$ is the absolute minimum of f , while at low temperature f has two equal minima $f(m_+) = f(m_-)$. In this line of reasoning, we are implicitly assuming that the external field is *exactly* 0 when we take the thermodynamic limit. However, this is not a satisfactory assumption: the magnetic field is a physical parameter, while the thermodynamic limit is an idealization, so that the description of the physical ferromagnet should be obtained by considering a finite size system in the presence of a (possibly small) magnetic field, and computing the thermodynamic limit of the system in the presence of the field, which can then be taken to 0. The expected magnetization in the absence of field is then

$$m_0 = \lim_{h^{\text{ext}} \rightarrow 0} \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E}[M | \beta, h^{\text{ext}}]. \quad (1.20)$$

As a consequence, the degeneracy between the two minima of f present when $\beta > 1$ is removed before we take the limit of zero field, and only one of the two minima will contribute to the Gibbs measure. Loosely speaking, in the presence of spontaneous magnetization $m_0 > 0$, in order to reach a configuration of magnetization $m < 0$ the system must cross a free energy barrier of order $O(N)$, which cannot occur in the thermodynamic limit: the configuration space then breaks in two distinct regions, one containing all the configurations with positive magnetization and the other those with a negative one, and the two regions are dynamically disconnected. This is an example of *ergodicity breaking* (for a clarifying discussion of ergodicity breaking in magnetic systems, see Chapter 2 of [3]).

A final remark concerning the nature of the phase transition. We can compute the magnetization as a function of the external field by looking at the positions of the minima of f . In the absence of field, when $\beta = 1 + \epsilon$ the two minima are separated by a distance of order $o(1)$ (as $\epsilon \rightarrow 0$), and the value of the spontaneous magnetization grows continuously from 0 to a finite value with $\beta - 1$. However, a different situation can occur, in which at the critical temperature the free energy has two *well separated* minima, such that one is favored for $\beta = \beta_c + \epsilon$ and the other for $\beta = \beta_c - \epsilon$. In this case, when the temperature crosses the critical value, the order parameter undergoes a discontinuous change. This kind of discontinuous phase transitions is called of *first order*, while continuous ones are called of *second order*.

1.2 Disordered systems and spin glasses

Disorder is ubiquitous in nature: amorphous materials are infinitely more common than crystals; biological systems sometimes manifest order in the form of regular behavior, but rarely of structure; the distribution of matter in the universe is irregular at any scale... Countless more examples show that, in fact, disorder is the rule of nature, and order is the exception.

However, the apparent lack of order and structure is not a sufficient criterion to consider a system as properly *disordered*. After all, a snapshot of the positions of molecules in a gas shows no sign of order, and yet gasses have a perfectly regular behavior under most conditions. On the other hand, a system as simple as a double pendulum can have an incredibly complicated dynamical evolution, with no signs of regularity at all, but would hardly be considered disordered.

In this section I shall try to give some examples of systems in which disorder plays a crucial role in determining their behavior, and which can be understood in terms of some very general concepts, in order to obtain a better characterization of what “proper” disordered systems are. I shall also introduce a formalism that has proven extremely powerful to describe them in a quantitative way.

1.2.1 Origins of disorder

In general, a disordered system can be characterized as having two distinct sets of parameters. The first one corresponds to the degrees of freedom of the system that have a dynamical evolution during the observation of the system. The second set corresponds to some parameters that influence the dynamics of the degrees of freedom, but that do not change during the observation, and which have “random” or irregular values.

In some cases the distinction between the two sets of variables will be purely dynamical. Glasses are a prototypical example of this kind of systems. They lack any long-range order, but locally the positions of atoms are very constrained. As a result, the motion of an atom typically requires the rearrangement of a number of neighbors that varies widely, and some degrees of freedom are effectively “frozen” over the experimental time scales, while others undergo a fast dynamical evolution. Another

example of this class of system is provided by kinetically constrained models, which are a simplification and generalization of glasses. These models generally study particles on lattices that undergo some simple dynamics, e.g. each site can be either empty or occupied by one particle, and particles can hop from one site to the next under some conditions that are specific to the model and which typically include that the site be empty. Depending on the boundary conditions and on the specific dynamical rules a rich phenomenology can be produced.

In other cases the distinction between dynamical variables and “frozen” parameters is explicit: some parameters (e.g. the interaction strength between pairs of particles) take constant random values, extracted from some known distribution. This kind of disorder is said to be *quenched*². The most celebrated example is that of magnetic impurities diluted in noble metal alloys, in which the positions of the impurities, and therefore the strengths of their magnetic interactions, are in fact random, giving rise to a very peculiar phenomenology. The theoretical models introduced to study these materials and to reproduce their behavior go under the name of spin glasses. The rest of this section will be devoted to introduce the most widely studied models of spin glasses, while their phenomenology and the analytical techniques used to solve them will be discussed in the latter sections of this Chapter.

1.2.2 Spin glass models

The simplest models for spin glasses has the following hamiltonian (for the classical introduction to the field, see [1]):

$$H_J = \sum_{i,j} J_{ij} \sigma_i \sigma_j \quad (1.21)$$

where the $J \equiv \{J_{ij}\}$ are random couplings and $\sigma \equiv \{\sigma_i\}$ are Ising spins. Depending on the geometry of the interaction, several models can be obtained:

Edwards-Anderson (EA) — The interactions involve only nearest neighbors on a lattice of dimension D , and their strengths are random variables extracted from a Gaussian distribution with zero average and finite variance. This was the first model introduced to describe magnetic alloys [4].

Sherrington-Kirkpatrick (SK) — Each J_{ij} (for each distinct couple of indices) is extracted from a Gaussian distribution. In order for the energy to be extensive, the standard deviation of the distribution must be of order $O(N^{-1/2})$ [5].

Bethe lattice — The interactions between spins are described by a Bethe lattice (i.e. a random graph with a finite connectivity k and with no loops), and their strength has a standard deviation proportional to $k^{-1/2}$.

A simple generalization is obtained by allowing the interaction to involve a number of spins $p > 2$:

$$H_J = \sum_{i_1, i_2, \dots, i_p} J_{i_1 i_2 \dots i_p} \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_p} . \quad (1.22)$$

In such **p-spin** models the spins can be either Ising or real ($\sigma_i \in \mathbb{R}$). In the latter case a spherical constraint $\sum_i \sigma_i^2 = 1$ is imposed. Many more models have been proposed and studied, which I shall not describe.

²Notice, however, that there is no fundamental difference between quenched and dynamically induced disorder: in both cases, a large number of parameters is effectively frozen in random values. The difference is mainly related to the description, rather than the physics of the system.

1.2.3 Mean field theory and diluted models

Even though the Edwards-Anderson model was the first spin glass model to be proposed, in 1975, it still waits for a general solution. In fact, most of the progress made in spin glasses has been obtained on the basis of *mean field theory*. Mean field theory can be defined as in the case of the Ising ferromagnet by writing the hamiltonian (1.21) in terms of local fields,

$$H_J(\sigma) = \sum_i h_i(\sigma) \sigma_i, \quad h_i(\sigma) = \sum_j J_{ij} \sigma_j \quad (1.23)$$

and replacing the configuration-dependent value of h_i with its thermal average, which depends on magnetizations rather than spin values. This approach can be generalized (and made much more powerful) by writing directly an expression for the free energy which depends on the local magnetizations $\{m_i\}$ and looking for the values of $\{m_i\}$ that satisfy the set of equations $\partial f / \partial m_i = 0$, an approach that goes under the names of Thouless, Anderson and Palmer (TAP) [6]. However much care should be exercised in deriving the expression for the free energy, and it should be kept in mind that since this doesn't (usually) come from a variational principle, there is no requirement for the solutions to the TAP equations to be minima of the free energy. As we shall see, the mean field results can be derived in a more transparent, but more complicated, analytical way.

A very important point to stress is that mean field results are in general exact for infinite range models, such as SK (and this has been recently rigorously proved), but are only approximations for large (but finite) range models, which become poor approximations if the range of interaction is short. This is due to the fact that in long range models, local fluctuations of thermodynamic quantities have no global effects, while in short range models they become crucial. However, finite range models have proven themselves very elusive so far. This raises the question of how to include local fluctuation effects in more tractable models.

A step towards this direction is provided by *diluted* models, of which the Bethe lattice model introduced in the previous subsection is an example. A more general case is obtained when the geometry of the model is an Erdős-Rényi random graph, in which each pair of spins has the same probability of being connected, and the average connectivity is finite. In these models, the corrections to mean field theory arise from loops, which are typically of length $O(\log N)$, and their magnitude is small and can be dealt with (as we shall see when I will introduce the cavity method). On the other hand, local fluctuations are present in diluted models, and they can be studied in this context.

1.2.4 Frustration, local degeneracies, complexity

A very general and important feature of the spin glass hamiltonian (1.21) is that its global minima, which govern the low temperature behavior of the system, cannot be found by *local* optimization. This fact has two causes, and very deep implications.

The first cause is *frustration*, which can be most simply illustrated by an example: if $J_{12}, J_{13} > 0$ while $J_{23} < 0$ there is no possible assignment of $\sigma_1, \sigma_2, \sigma_3$ that will make all three terms in $J_{12}\sigma_1\sigma_2 + J_{13}\sigma_1\sigma_3 + J_{23}\sigma_2\sigma_3$ negative. Some of the addends in the hamiltonian will have to be positive, and the minimization of the hamiltonian requires a global approach.

Also, once it is clear that some interactions will have to give positive contributions, it is also clear that a large number of choices are possible for *which* terms to make positive: in general a large number of configurations will have the ground state energy density. But this local degeneracy, which is the second obstacle to local optimization, can occur independently of frustration. If we consider (only for the sake of this argument) an Ising p -spin model with large p and all the J 's positive, we see that

the number of assignments that minimize each term in the hamiltonian (separately) is 2^{p-1} . Each many-spin interaction term poses a very weak constraint on the individual spins.

The consequence of frustration and local degeneracy is that in general the ground state of a spin glass will be highly degenerate. Not only the number of minimal energy configurations will be exponential in the size N of the system, but often, due to disorder, the Gibbs measure will decompose in a large number \mathcal{N} of pure states. In some cases this number will be exponential: $\mathcal{N} \sim e^{\Sigma N}$ where $\Sigma > 0$ is called complexity; in other cases \mathcal{N} will be sub-exponential in N , but still large.

1.2.5 The order parameter of disordered systems

The most striking feature of spin glasses is that there is order hidden in their disorder. If one looks at a “typical” configuration of a spin glass, it will look the same at any temperature: each spin points in an apparently random direction. However, as the temperature is lowered, each spin becomes more and more “frozen” in a particular direction, which will depend on the site and which will “look” as disordered as the typical high temperature configuration. At sufficiently low temperatures, even though the site-averaged magnetization is zero, the local average magnetization is not. A convenient measure of this hidden order was introduced by Edwards and Anderson [4], and goes under their names:

$$q_{\text{EA}} = \frac{1}{N} \sum_i m_i^2 \quad (1.24)$$

where m_i is the thermal average of σ_i . In the following I shall denote thermal averages with angled brackets, e.g. $m_i = \langle \sigma_i \rangle$.

Of course, since the hamiltonian is dependent on the specific values of the random couplings, the value of m_i will also depend on them. However, for many physical observables the average over sites is equal to the average over disorder:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_i O_J(i) = \overline{O_J(\cdot)} \equiv \int d\mu(J) O_J(\cdot) \quad (1.25)$$

where $\mu(J)$ is the distribution of disorder. Such observables are said to be *self-averaging*, and the Edwards-Anderson order parameter q_{EA} is one of them. On the other hand, if physically relevant observables were to be dependent on the realization of disorder, i.e. on the specific sample, there would be very little to say about them, and very little interest in their study.

The Edwards-Anderson order parameter is very closely related to a more general quantity, the overlap, which can be defined on two different contexts. The overlap between microscopic configurations σ and τ can be defined as

$$q_{\sigma\tau} = \frac{1}{N} \sum_i \sigma_i \tau_i \quad (1.26)$$

which will be in the interval $[-1, 1]$. The value 1 will correspond to perfectly correlated configurations, -1 to perfectly anti-correlated ones, and 0 to uncorrelated σ and τ . The concept of overlap can be extended to thermodynamic states, and is particularly interesting in the presence of ergodicity breaking. If we consider two different thermodynamic states α and β , we can compute

$$q_{\alpha\beta} = \frac{1}{N} \sum_i \langle \sigma_i \rangle_\alpha \langle \sigma_i \rangle_\beta \quad (1.27)$$

which will measure how different the two states are.

When a single state is present, the Edwards-Anderson order parameter is just $q_{\text{EA}} = q_{\alpha\alpha}$, the self-overlap of the state with itself. However, in presence of ergodicity breaking, the Gibbs measure decomposes in a sum over pure states,

$$\langle O \rangle = \sum_{\sigma} \frac{1}{Z} O(\sigma) e^{-\beta H(\sigma)} = \sum_{\alpha} \frac{Z_{\alpha}}{Z} \sum_{\sigma \in \alpha} \frac{1}{Z_{\alpha}} O(\sigma) e^{-\beta H(\sigma)} = \sum_{\alpha} w_{\alpha} \langle O \rangle_{\alpha} \quad (1.28)$$

where $Z_{\alpha} \equiv \sum_{\sigma \in \alpha} \exp(-\beta H(\sigma))$ and $w_{\alpha} \equiv Z_{\alpha}/Z$ is the relative weight of the state α in the decomposition. In this case, the Edwards-Anderson parameter is given by

$$q_{\text{EA}} = \overline{\frac{1}{N} \sum_i \langle \sigma_i \rangle^2} = \overline{\frac{1}{N} \sum_i \left(\sum_{\alpha} w_{\alpha} \langle \sigma_i \rangle_{\alpha} \right)^2} = \overline{\sum_{\alpha, \beta} w_{\alpha} w_{\beta} q_{\alpha\beta}} \quad (1.29)$$

in which not just the self-overlaps of the states are considered, but also the overlaps among different states.

A very powerful characterization of the structure of the thermodynamic states is provided by the distribution of overlaps between states,

$$\mathcal{P}(q) = \overline{\sum_{\alpha, \beta} w_{\alpha} w_{\beta} \delta(q - q_{\alpha\beta})} \quad (1.30)$$

which gives the probability that two configurations picked at random from the Gibbs distribution have overlap q . In terms of $\mathcal{P}(q)$ we will have

$$q_{\text{EA}} = \int dq \mathcal{P}(q) q. \quad (1.31)$$

1.3 Phenomenology of disordered systems

As I have tried to explain in the previous section, disordered systems share three characteristic features: first, the presence of quenched disorder; second, the effects of frustration and local degeneracy, which lead to the existence of many thermodynamic states at low temperature; third, the “freezing” of the dynamical degrees of freedom in a disordered configuration at low temperature. From the phenomenological point of view, the two latter characteristics are the most relevant ones.

In this section I shall briefly review the phenomenology of disordered systems that support this picture, and which is common to a very wide class of systems, regardless of the specificities of different models.

1.3.1 Spin glass susceptibilities

The first clear observation of a “hidden” order in disordered systems came from measures of the low-field AC magnetic susceptibility in diluted solutions of iron in gold. The magnetic susceptibility χ is directly related to the Edwards-Anderson order parameter q_{EA} . It is defined locally as $\chi_{ii} = \partial m_i / \partial h_i^{\text{ext}}$, where h_i^{ext} is the applied external field. Since the contribution of the external field to the hamiltonian is always a linear term $-\sum_i h_i^{\text{ext}} \sigma_i$, it is easy to see that the following fluctuation-response relation must hold:

$$\chi_{ii} = \frac{\partial m_i}{\partial h_i^{\text{ext}}} = \frac{\partial^2}{\partial (h_i^{\text{ext}})^2} \frac{1}{\beta} \log Z(\beta, \{h_i^{\text{ext}}\}) = \beta \left\langle (\sigma_i - \langle \sigma_i \rangle)^2 \right\rangle = \beta(1 - m_i^2). \quad (1.32)$$

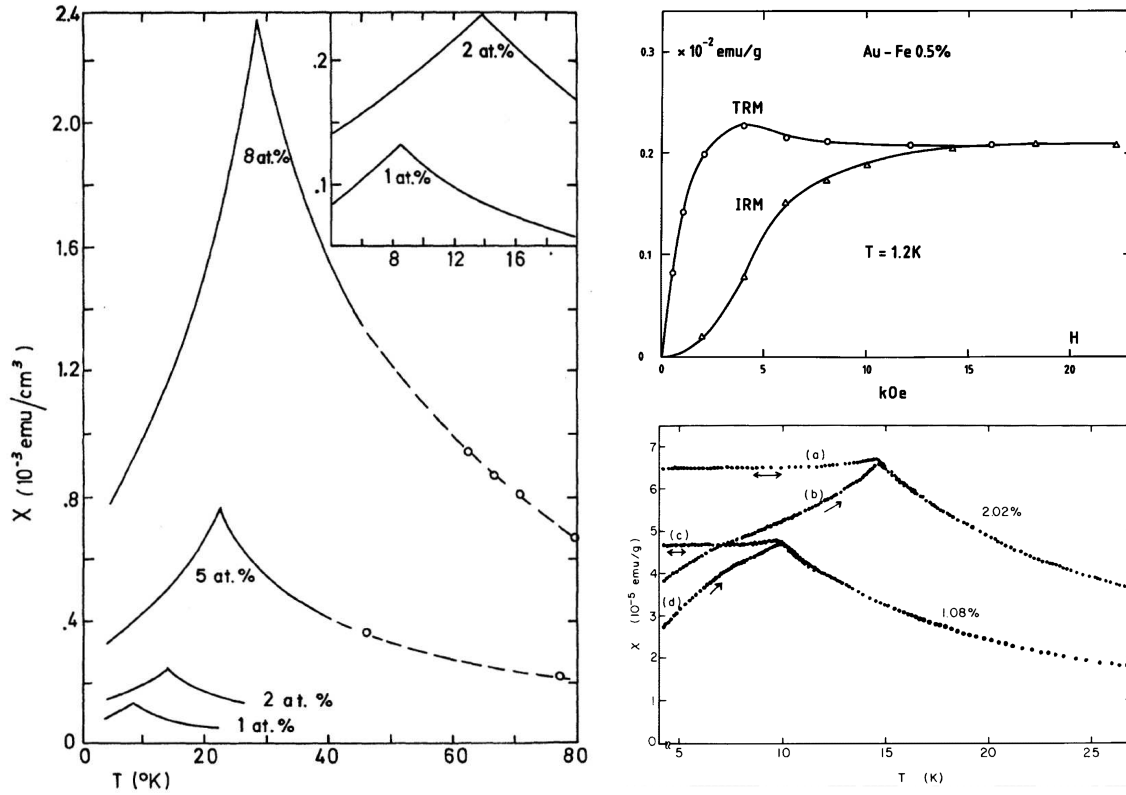


Figure 1.1: Magnetic properties of spin glasses. *Left* The AC susceptibility of AuFe alloys at different Fe concentrations for low field ($\simeq 5$ G) and $\nu = 155$ Hz (from [7]). *Right bottom* The DC susceptibility of CuMn for two Mn concentrations. Curves (a) and (c) were obtained by cooling in the measurement field (FC), (b) and (d) are the results of zero-field-cooled (ZFC) experiments (from [8]). *Right top* Remanent magnetization in AuFe (from [9]).

The measured local susceptibility is the average of χ_{ii} over the sites:

$$\chi_{\text{loc}} = \frac{1}{N} \sum_i \chi_{ii} = \beta(1 - q_{\text{EA}}). \quad (1.33)$$

In the absence of magnetic ordering at low temperatures, χ_{loc} should diverge as $1/T$. The measured susceptibility shows a sharp cusp instead of a divergence, which indicates that below a certain temperature $q_{\text{EA}} > 0$ (Fig. 1.1).

A more detailed analysis of the frequency dependence of the measured AC susceptibility suggests the existence of a *glassy* magnetic phase, i.e. a phase characterized by the existence of many metastable states. This is clearly confirmed by measures of DC magnetic susceptibility and of remanent magnetization, which both display a very strong dependence of the response on the details of the preparation of the sample. In DC susceptibility measures it can be seen that below a critical temperature, which coincides with the extrapolation to zero frequency of the position of the cusps in AC measurements, two different values of susceptibility can be measured: if the sample is cooled in the absence of field one obtains χ_{zfc} , which is lower than χ_{fc} , the value which is obtained when the sample is cooled in the presence of field. Moreover, if the external field is strong, a “remanent” magnetization is observed after it is switched off. The value of the remanent magnetization again

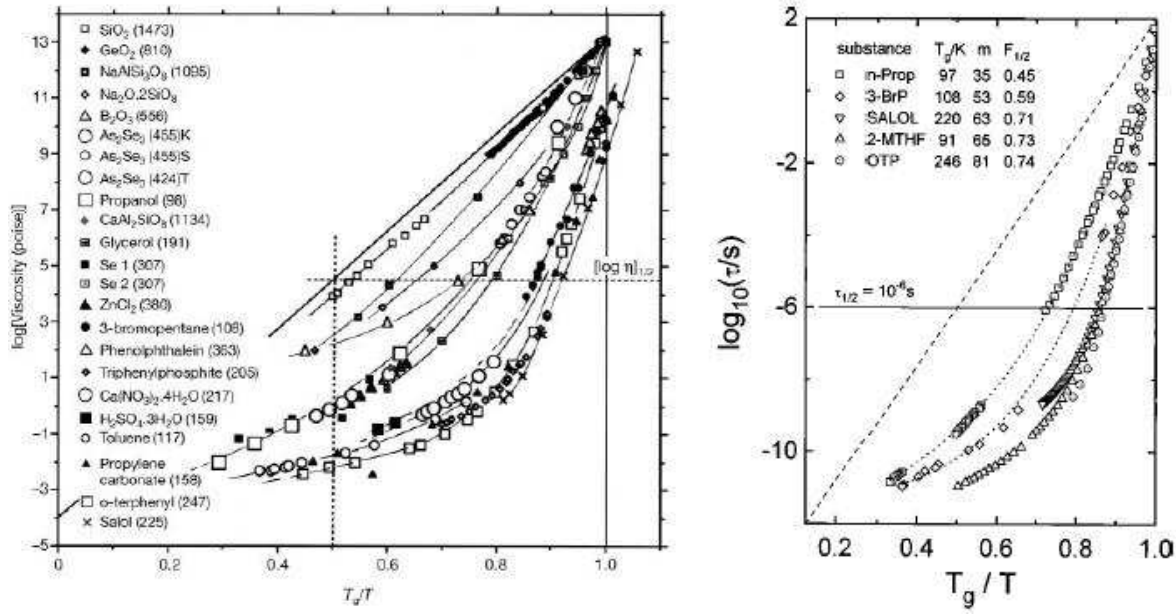


Figure 1.2: *Left* Viscosity measures for many glass forming liquids (from [10]). The glass forming temperature T_g is reported in parenthesis in the legend for each liquid. *Right* Structural relaxation times from dielectric relaxation measurements (from [11])

depends on whether the field was applied during the cooling of the sample or only later. In the first case, the so called Thermo-Rmanent Magnetization (TRM) is larger than the Isothermal Rmanent Magnetization (IRM) (Fig. 1.1). This dependence on preparation of the sample properties clearly demonstrate that many different low temperature thermodynamic states are accessible to the system, and that they are well separated from each other, in the sense that the free energy barriers between states are extensive.

1.3.2 Divergence of relaxation times

The main characteristic of glassy behavior is the divergence of the relaxation time at finite temperature. For structural glasses, the relaxation time t_α is defined as the decay time of density fluctuations, and it is accessible experimentally both directly and through the Maxwell relation

$$\eta = G_\infty t_\alpha \quad (1.34)$$

where η is the viscosity and G_∞ is the infinite-frequency shear modulus of the liquid. Experiments show that super-cooled liquids have a viscosity which can vary by as much as 15 orders of magnitude when the temperature varies by a factor of two above the glass forming temperature (Fig. 1.2). Similar results are obtained from direct measurements.

Spin glass models also show a divergence in relaxation times. A good example is provided by the p -spin spherical model (for $p \geq 3$). At high temperatures, the Fluctuation-Dissipation Theorem (FDT) holds, and the correlation $C(t, t')$ is related to the response $F(t, t')$ by the relation

$$\frac{\partial}{\partial t} C(t, t') = -T F(t, t'). \quad (1.35)$$

If the system equilibrates, the correlation function becomes invariant under time translations, $C(t, t +$

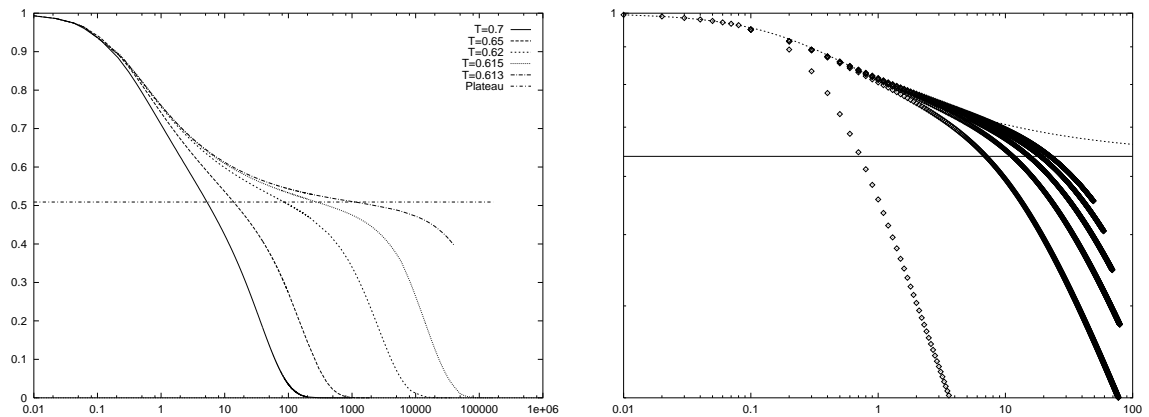


Figure 1.3: *Left* The translationally invariant correlation function $C_{\text{eq}}(\tau)$ as a function of τ , for different temperatures T . The horizontal line is the value of q_{EA} . *Right* The out of equilibrium correlation function $C(t_w, t_w + t)$ as a function of t for different waiting times t_w at temperature $T = 0.5$. The dotted line is computed in the limit $t_w \rightarrow \infty$ and the horizontal line is its limiting value for $t \rightarrow \infty$. Both figures are from [14].

$\tau) = C_{\text{eq}}(\tau)$ and it is possible to derive a differential equation for $C_{\text{eq}}(\tau)$, whose numerical solution for $p = 3$ is shown in figure 1.3.

What one sees is that as the temperature is decreased, a *plateau* forms. The length of the plateau diverges as $T \rightarrow T_d$. The analysis of the model shows that T_d is the temperature at which the free energy becomes dominated by an exponential number of metastable states with energy higher than the ground state. The value of the plateau coincides with q_{EA} .

1.3.3 Ageing

If the temperature is lowered below T_d , a striking break-down of the translational invariance of the correlation function occurs, signalling that the system becomes unable to equilibrate. In this regime, the correlation function $C(t_w, t_w + t)$ depends separately on the waiting time t_w and on the duration of the observation t . Only in the limit $t_w \rightarrow \infty$ the validity of the FDT is recovered and the system finally equilibrates.

This is an example of a very general phenomenon, observed in structural glasses as well as in spin glasses, which goes under the name of *ageing*. Many observables for disordered systems maintain a time dependence for very long times under stable external conditions, indicating that they cannot equilibrate. This again confirms the existence of many metastable states which “trap” the dynamics of the system.

1.4 The replica method

In this section I shall briefly review one of the two equivalent analytical methods that can be used to investigate the equilibrium properties disordered systems: the replica method [1].

1.4.1 The replica trick

As I mentioned in the second section of this chapter, many physically relevant quantities are self-averaging, which is to say that their thermodynamic average is independent on the specific sample.

A most notable example of a self-averaging quantity is the free energy density,

$$f_J(\beta) = - \lim_{N \rightarrow \infty} \frac{1}{\beta N} \log Z_J(\beta) \quad (1.36)$$

where the subscript J denotes the dependence on the disorder. Because of the self-averageness of f , the free energy of any sample will be the same, and will be equal to the average over the distribution of J of f_J :

$$f(\beta) = \overline{f_J(\beta)} = - \lim_{N \rightarrow \infty} \frac{1}{\beta N} \overline{\log Z_J(\beta)} \equiv - \lim_{N \rightarrow \infty} \frac{1}{\beta N} \int d\mu(J) \log \sum_{\sigma} e^{-\beta H_J(\sigma)}. \quad (1.37)$$

Unfortunately, the presence of the logarithm in the integral over the disorder makes it impossible to calculate it directly. However, one can use the following identity

$$\log x = \lim_{n \rightarrow 0} \frac{x^n - 1}{n} \quad (1.38)$$

and write

$$\overline{\log Z_J(\beta)} = \overline{\lim_{n \rightarrow 0} \frac{Z_J(\beta)^n - 1}{n}} = \lim_{n \rightarrow 0} \frac{\overline{Z_J(\beta)^n} - 1}{n} = \lim_{n \rightarrow 0} \log \overline{Z_J(\beta)^n}. \quad (1.39)$$

By doing this, instead of $\overline{\log Z_J(\beta)}$ one has to compute $\overline{Z_J(\beta)^n}$, which turns out to be much simpler. Notice that $Z_J(\beta)^n$ is the partition function of a system in which the dynamical degrees of freedom are replicated n times and the quenched parameters are the same in each replica (hence the name, *replica trick*).

1.4.2 Solution of the p -spin spherical model

As an example of the replica method, I am going to sketch its application to the p -spin spherical model. The hamiltonian is given by

$$H_J(\sigma) = \sum_{i_1, \dots, i_p} J_{i_1 \dots i_p} \sigma_{i_1} \cdots \sigma_{i_p}. \quad (1.40)$$

The disorder J has a gaussian distribution with average 0, and in order for the hamiltonian to be extensive its variance must scale as N^{p-1} :

$$\mathbb{P}[J_{i_1 \dots i_p} = J] = \mu(J) = \sqrt{\frac{2N^{p-1}}{2\pi p!}} \exp \left\{ -\frac{1}{2} J^2 \frac{2N^{p-1}}{p!} \right\}. \quad (1.41)$$

The starting point is to compute the gaussian integral over the gaussian distribution of disorder:

$$\overline{Z_J(\beta)^n} = \prod_{i_1 \dots i_p} \int d\mu(J_{i_1 \dots i_p}) \int d\sigma^1 \cdots d\sigma^n \exp \left\{ -\beta J_{i_1 \dots i_p} \sum_{a=1}^n \sigma_{i_1}^a \cdots \sigma_{i_p}^a \right\} \quad (1.42)$$

$$= \prod_{i_1 \dots i_p} \int d\sigma^1 \cdots d\sigma^n \exp \left\{ \frac{\beta^2 p!}{4N^{p-1}} \sum_{a,b}^{1,n} \sigma_{i_1}^a \sigma_{i_1}^b \cdots \sigma_{i_p}^a \sigma_{i_p}^b \right\} \quad (1.43)$$

where I have dropped an overall normalization constant which doesn't give an extensive contribution. Here and in the following, I shall always denote by i, j, k, \dots site indices running from 1 to N and with a, b, c, \dots replica indices running from 1 to n . Notice that after the integral we are left with a system in which *sites are independent* and *replicas are coupled*, which we can rewrite:

$$\overline{Z_J(\beta)^n} = \int d\sigma^1 \cdots d\sigma^n \exp \left\{ \frac{\beta^2}{4N^{p-1}} \sum_{a,b}^{1,n} \left(\sum_i \sigma_i^a \sigma_i^b \right)^p \right\} \quad (1.44)$$

We can now introduce the overlaps between replicas,

$$Q_{ab} = \frac{1}{N} \sum_i \sigma_i^a \sigma_i^b \quad (1.45)$$

and multiply by

$$1 = \int dQ_{ab} \int d\lambda_{ab} \exp \left\{ i\lambda_{ab} \left[NQ_{ab} - \sum_i \sigma_i^a \sigma_i^b \right] \right\} \quad (1.46)$$

to obtain:

$$\overline{Z_J(\beta)^n} = \int d\sigma^a \cdots d\sigma^n \int dQ \int d\lambda \exp \left\{ \frac{\beta^2 N}{4} \sum_{ab} Q_{ab}^p + N \sum_{ab} \lambda_{ab} Q_{ab} - \sum_i \sum_{ab} \sigma_i^a \lambda_{ab} \sigma_i^b \right\} \quad (1.47)$$

(where $Q \equiv \{Q_{ab}\}$ and $\lambda \equiv \{\lambda_{ab}\}$). This integral is now gaussian in σ , and can be performed to obtain:

$$\overline{Z_J(\beta)^n} = \int dQ d\lambda e^{-NS(Q,\lambda)} \quad (1.48)$$

where the action is

$$S(Q, \lambda) = -\frac{\beta^2}{4} \sum_{ab} Q_{ab}^p - \sum_{ab} \lambda_{ab} Q_{ab} + \frac{1}{2} \log \det(2\lambda). \quad (1.49)$$

This integral can be done using the Laplace method, in order to obtain

$$f = -\lim_{N \rightarrow \infty} \frac{1}{\beta N} \lim_{n \rightarrow 0} \frac{1}{n} \log \int dQ d\lambda e^{-NS(Q,\lambda)} = -\lim_{n \rightarrow 0} \frac{1}{n} \lim_{N \rightarrow \infty} \frac{1}{\beta N} \log e^{-NS(\bar{Q}, \bar{\lambda})} = \lim_{n \rightarrow 0} \frac{1}{\beta n} S(\bar{Q}, \bar{\lambda}) \quad (1.50)$$

where \bar{Q} and $\bar{\lambda}$ extremize the action. Notice however that we had to invert the order in which the limits over N and n are taken, which is not *a priori* a legitimate manipulation. Assuming it to be correct, the saddle point equations one obtains are the following:

$$\lambda_{ab} = \frac{1}{2} (Q^{-1})_{ab}, \quad (1.51)$$

$$\frac{\partial f}{\partial Q_{ab}} = 0 = \frac{\beta^2 p}{2} Q_{ab}^{p-1} + (Q^{-1})_{ab} \quad (1.52)$$

As we see, the parameter space over which one has to minimize f is the space of symmetric matrices Q . The dimension of these matrices is n , which is assumed to go to 0: the only way to obtain a meaningful result is to write an expression for f which is valid for *any* finite n and then do an analytic continuation of this expression for $n \rightarrow 0$. However, this requires that the matrix Q_{ab} be parameterized in such a way that the matrix elements will depend on n and on a fixed number r of parameters $\{p_1, p_2, \dots, p_r\}$, which will be set to the values that satisfy the saddle point equations and which will be functions of n .

This rather intricate procedure raises three issues. The first is related to the fact that the whole procedure is far from rigorous from the mathematical point of view. Second, the parameterization of Q in the particular form I've described limits the scope for the extremalization of f : it is not at all clear *a priori* that the absolute extremum of f corresponds to a matrix of the “right” form, and we may end up with an extremum that is not the “true” one. Finally, the stability of the free energy which is obtained in the end should be carefully checked *a posteriori*. I shall return on these issues later.

A “naive” hypothesis would be to assume that since the replicas are just a formal expedient to compute f , the physical quantities should be independent of the replica index, and the overlap matrix

Q should be invariant under permutations of the replica indices. This would lead to the very simple parameterization $Q_{ab} = q_0 + (1 - q_0)\delta_{ab}$ (the diagonal elements of Q are determined by the spherical constraint to be 1). However, as already noted, the replicas *do* have a physical interpretation: the replicated partition function, which is the proper self-averaging quantity to compute, corresponds to a composite system consisting of n replicas of the original one. There is no reason why, in the presence of many states, different replicas should find themselves in the same state. Quite on the contrary, one should expect the breaking of the replica symmetry to be the signature of the presence of many states. It turns out that this intuition is correct.

The solution of the p -spin model [12] can be obtained by applying the replica-symmetry breaking (RSB) scheme introduced by Parisi to solve the SK model [15, 16, 17]. The following parameterization is assumed for Q :

$$Q_{ab} = \delta_{ab} + q_1(1 - \delta_{ab})\mathbb{I}(a \div m = b \div m) + q_0\mathbb{I}(a \div m \neq b \div m) \quad (1.53)$$

where the free parameters are $\{m, q_0, q_1\}$, $a \div m$ represents the integer division of a by m , and $\mathbb{I}(\text{event})$ is the indicator function of *event* (i.e. it is 1 if *event* is true and 0 otherwise). The parameters are subject to the conditions $0 < m < n$, with m such that n is a multiple of m , and $0 < q_0 < q_1 < 1$. This parameterization corresponds to a matrix Q which is made of n/m identical blocks of size m covering the main diagonal, with 1 on the main diagonal and q_1 outside of it in each block, and q_0 outside the blocks (notice that the case $m = n$ and $q_0 = q_1$ would correspond to the replica symmetric solution). This parameterization is known as *one-step replica-symmetry breaking*, or 1RSB for short.

If this parameterization is substituted in the expression of the action $S(Q, \lambda)$ (1.49), and the limit $n \rightarrow 0$ is computed, the following expression for the free energy is obtained:

$$\begin{aligned} f_{\text{1RSB}} = & -\frac{\beta}{4}[1 + (m-1)q_1^p - mq_0^p] - \frac{1}{2\beta} \left\{ \frac{m-1}{m} \log(1 - q_1) + \right. \\ & \left. + \frac{1}{m} \log[m(q_1 - q_0) + (1 - q_1)] + \frac{q_0}{m(q_1 - q_0) - (1 - q_1)} \right\}. \end{aligned} \quad (1.54)$$

This expression can then be minimized to obtain the values of m , q_1 and q_0 . What one sees is that for high temperature, a solution with $m = 1$ exists and is stable. However (for $p \geq 3$) as the temperature is lowered to T_s the solution with $m = 1$ becomes *unstable* and a new solution with $m < 1$ appears, which is stable and has a lower free energy than the solution with $m = 1$. The value of m undergoes a discontinuity as T crosses T_s , jumping from 1 to a value which is at a finite distance from 1. As I have already mentioned, the existence of a replica-symmetry breaking solution is the signature of a glassy phase in which many different thermodynamic states coexist. The p -spin model undergoes a phase transition at T_s from a paramagnetic to a glassy phase.

I would like to conclude this section with three remarks. The first concerns the issues I mentioned regarding the validity of the replica method. As I wrote, in general the procedure is not mathematically rigorous. However, one should note that in the case of the SK model the Parisi solution has been recently proved to be exact. Moreover the method has been applied to a large number of fairly different models, and in each case the results obtained are sensible: it appears safe to conjecture its validity, with the proviso that the stability of the solution it gives should be checked *a posteriori* and that one cannot rule out the existence of other solutions, possibly with lower free energy.

Second, the example of the p -spin is particularly simple. In other models, including SK, one needs to consider a more complicated parameterization of the overlap matrix, which consists in applying the procedure I described recursively: one starts with a “block” of size m_i which has 1 on the main diagonal and q_i outside of it, and introduces blocks of size $m_{i+1} = m_i \div p$ (for some integer p) on

the diagonal, with the same structure as the starting block, but a new value q_{i+1} for the off-diagonal elements. This procedure can be repeated for any number of *steps*. The solution of the p -spin is *one-step replica-symmetry breaking*, denoted 1RSB. In the case of the SK model one needs an infinite number of steps, and the solution is said to be *full replica-symmetry breaking* (FRSB).

Finally, the parameters over which one needs to extremize the free energy are the matrix elements of Q (through the Parisi parameterization), which are *scalar* quantities. This is a general feature of fully-connected models. However, as we shall see in the following section, the parameters to be minimized become much more complicated in the case of diluted models.

1.4.3 Replica formalism for diluted models

In order to apply the replica method to diluted systems, one needs to generalize the approach that I have outlined for the case of the p -spin [19, 20, 21, 22]. The starting point is the same: the average over disorder of the n -replicated partition function. For a system of Ising spins $\sigma_i \in \{-1, 1\}$, with $\sigma \equiv \{\sigma_1, \dots, \sigma_N\}$ and with hamiltonian $H_J(\sigma)$, we have:

$$\overline{Z_J(\beta)^n} = \int d\mu(J) \sum_{\sigma^1} \cdots \sum_{\sigma^n} \exp \left\{ -\beta \sum_{a=1}^n H_J(\sigma^a) \right\} = \sum_{\sigma^1} \cdots \sum_{\sigma^n} \overline{\exp \left\{ -\beta \sum_{a=1}^n H_J(\sigma^a) \right\}} \quad (1.55)$$

where σ^a is the N -spin configuration of the a^{th} replica. In fact, the n -replicated spin configuration is a matrix σ with N rows corresponding to the sites and n columns corresponding to the replicas. The i^{th} row is the n -component vector $\vec{\sigma}_i$ in which the component σ_i^a is the value of the spin on the site i for replica a , and the a^{th} column is the N -component configuration of replica a .

As an example of hamiltonian, we can consider the diluted version of the Ising p -spin model, which we shall discuss more in detail in the following:

$$H_J(\sigma) = \sum_{m=1}^M \frac{1}{2} \left(1 - J_m \sigma_{i_1^m} \cdots \sigma_{i_p^m} \right) \quad (1.56)$$

where the sum is over M terms, each consisting of the product of p spin, with indices i_j^m with $j = 1, \dots, p$ selected uniformly at random between 1 and N , and where the couplings J_m are ± 1 uniformly at random. The additive constant present in each term of the sum is such that the energy is positive or null. The factor $1/2$ is such that the value of the energy is equal to the number of terms in the sum which have a J_m with a different sign relative to the product of the spins. On a random configuration, half the terms will be equal to 1 and the other half to 0, so that the energy will be extensive if $M = O(N)$.

We can interpret the right hand side of (1.55) as the partition function of an effective hamiltonian \mathcal{H} depending on the full replicated configuration σ :

$$\overline{Z_J(\beta)^n} = \sum_{\sigma} \exp \{ -\beta \mathcal{H}(\sigma) \} \quad (1.57)$$

Since the distribution of disorder is independent on the site, the averaged quantity in the right hand side of (1.55) must be invariant under permutations of site indices. This implies that the effective hamiltonian (1.57) can depend on σ only through

$$c(\vec{\tau}) \equiv \frac{1}{N} \sum_i \mathbb{I}(\vec{\sigma}_i = \vec{\tau}) \quad (1.58)$$

which is the fraction of sites that have replicated configuration $\vec{\tau}$. Even though $c(\vec{\tau})$ actually depends on the replicated configuration σ , we are going to assume it to be fixed and avoid its appearance in the notation. Also, notice that $\sum_{\vec{\tau}} c(\vec{\tau}) = 1$.

The overlap between replica configurations Q_{ab} can also be expressed in terms of $c(\sigma)$:

$$Q_{ab} = \frac{1}{N} \sum_i \sigma_i^a \sigma_i^b = \sum_{\vec{\tau}} c(\vec{\tau}) \tau^a \tau^b. \quad (1.59)$$

This was to be expected: in the calculation for the p -spin, the free energy we obtained depended only on Q , and (1.59) implies that what we obtained was actually dependent on $c(\vec{\tau})$ only. This is a general feature of fully connected models: their free energies (or rather, the actions whose extrema are equal to the free energy) depend only on the overlaps between replicas. However, for diluted models one needs to generalize (1.59) to include higher moments:

$$Q_{a_1 \dots a_k} \equiv \sum_{\vec{\tau}} c(\vec{\tau}) \tau^{a_1} \dots \tau^{a_k}. \quad (1.60)$$

The crucial point is that even though these quantities are more complicated than the overlaps, they are still conceptually equivalent to $c(\vec{\tau})$, which provides the full description of the structure of the states of the system, be it fully connected or diluted.

To see more in details how it is possible to write the free energy in terms of $c(\vec{\tau})$, we can go back to (1.57) where we recall that $\mathcal{H}(\sigma) = \mathcal{H}[c(\vec{\tau})]$:

$$\overline{Z_J(\beta)^n} = \sum_{\sigma} e^{-\beta \mathcal{H}(\sigma)} = \sum_{\{c(\vec{\tau})\}}^{0,N} \frac{N!}{\prod_{\vec{\tau}} [N c(\vec{\tau})]!} e^{-\beta \mathcal{H}[c(\vec{\tau})]} \mathbb{I} \left[\sum_{\vec{\tau}} c(\vec{\tau}) = 1 \right] \quad (1.61)$$

where the sum is over 2^n variables, each variable being the value of c for one of the possible 2^n n -component spin configurations, that take values between 0 and N , where the multinomial factor is just the number of replicated configurations σ that give rise to the same distribution $c(\vec{\tau})$, and where the last indicator function ensures the normalization of $c(\vec{\tau})$.

In the limit $N \rightarrow \infty$ the sum becomes an integral and the multinomial coefficient can be approximated with Stirling's formula to obtain

$$f(\beta) = - \lim_{N \rightarrow \infty} \frac{1}{\beta N} \lim_{n \rightarrow 0} \frac{1}{n} \overline{Z_J(\beta)^n} \quad (1.62)$$

$$= - \lim_{n \rightarrow 0} \frac{1}{n} \lim_{N \rightarrow \infty} \frac{1}{\beta N} \int_0^1 \left(\prod_{\vec{\tau}} dc(\vec{\tau}) \right) \exp \left\{ N \left[- \left(\sum_{\vec{\tau}} c(\vec{\tau}) \log c(\vec{\tau}) \right) - \beta \mathcal{H}[c(\vec{\tau})] \right] \right\} \times \\ \times \mathbb{I} \left[\sum_{\vec{\tau}} c(\vec{\tau}) = 1 \right] \quad (1.63)$$

$$= \lim_{n \rightarrow 0} \frac{1}{n} \frac{1}{\beta} \text{extremum}_{\{c(\vec{\tau}): \sum_{\vec{\tau}} c(\vec{\tau})=1\}} \left\{ \sum_{\vec{\tau}} c(\vec{\tau}) \log c(\vec{\tau}) + \beta \mathcal{H}[c(\vec{\tau})] \right\} \quad (1.64)$$

where (as before) we have exchanged the order of the limits $N \rightarrow \infty$ and $n \rightarrow 0$.

With this formalism, the problem of computing the free energy of a (possibly diluted) disordered Ising model is decomposed into three tasks:

1. Find the effective hamiltonian $\mathcal{H}[c(\vec{\tau})]$
2. Compute, for each value of n , the extremum of the free energy functional in $c(\vec{\tau})$ appearing on the right hand side of (1.64)

3. Perform the analytic continuation of the result to $n = 0$

In Chapter 5 I shall use this formalism to derive some properties of the solutions of an optimization problem which is formally equivalent to a diluted Ising spin glass.

Chapter 2

Optimization problems and algorithms

In the previous Chapter, I have given a very brief overview of the physics of disordered systems. In this Chapter, I shall introduce a different kind of disordered systems, which arise from the study of combinatorial optimization problems, and I shall discuss some aspects specific to them, and what they have in common with the disordered systems studied in physics.

In the first Section, I shall give some examples of combinatorial optimization problems; in Section 2.2 I shall introduce the two specific problems that have been the subject of my research, k -SAT and k -XORSAT; then I shall introduce some notions from complexity theory, in Section 2.3; finally, in 2.4 I shall present some families of algorithm that are useful for finding solutions to optimization problems, and whose properties also shed some light on the underlying structure of the problems themselves.

Most of the material discussed in this Chapter can be found in [2].

2.1 Some examples of combinatorial optimization problems

Optimization problems are concerned with finding the “best” (or *optimal*) allocation of finite resources to achieve some purpose. It is clearly a very general and important class of problems. An early example of optimization problem is narrated in Virgil’s Aeneid: Dido, a Phenician princess, is obliged to flee Tyre, her hometown, after her husband is murdered by her brother, a cruel tyrant. She embarks with a small group of refugees, and lands in Lybia, where she asks the king Iarbas to purchase some land to found a new city, Carthage. Iarbas, in love with Dido but rejected by her, has no intention to allow the settlement, and offers only as much land as can be enclosed in a bull’s hide. He is, however, outwitted by Dido, who cuts the hide in thin stripes, which she joins to form a long string. With that, she encloses an area shaped as a semi-circle, delimited by the sea, and sufficient to build Carthage. In this legendary tale, Dido not only had the brilliant idea of cutting the hide, but also solved a non-trivial optimization problem: what is the curve of given perimeter that encloses the largest area?

Combinatorial optimization problems are, in a way, simpler: the set of possible solutions is *discrete*. This restriction might appear severe in view of practical applications, but in fact it is not: many resources, such as industrial machines, skilled workers or computer chips are indeed indivisible. Let us begin with an example, which I shall use to illustrate a general, formal definition, and after which I shall give some more examples of different families of combinatorial optimization problems.

Consider the following

Knapsack Problem (KP) Given a set \mathcal{S} of items $i = 1, \dots, N$, each having a value $v_i \in \mathbb{R}^+$ and a weight $w_i \in \mathbb{R}^+$, what is the subset $\mathcal{S}' \subset \mathcal{S}$ with the largest total value $V = \sum_{i \in \mathcal{S}'} v_i$ and such that the total weight $W = \sum_{i \in \mathcal{S}'} w_i$ is $W \leq W^*$ for some given W^* ?

The possible solutions (or configurations) are all the subsets that can be formed with elements from \mathcal{S} , which are a discrete set of cardinality 2^N (corresponding to the two choices “present” or “not-present” for each item in \mathcal{S}). A specific instance of the general problem is defined by the pairs $\{(v_i, w_i), i = 1, \dots, N\}$, and by the maximum allowed weight W^* .

In general, an instance of the problems I shall consider will be defined by specifying the following three characteristics:

1. A set \mathcal{C} of possible *configurations* \mathcal{C} ;
2. A *cost function* $F : \mathcal{C} \rightarrow \mathbb{R}$ that associates a cost $F(\mathcal{C})$ to every configuration $\mathcal{C} \in \mathcal{C}$, and which can be computed in polynomial time;
3. An *objective*, that is to say a condition on $F(\mathcal{C})$ which must be satisfied.

In the knapsack example, \mathcal{C} is the set of all subsets of \mathcal{S} , the cost function F is

$$F(\mathcal{C}) = \mathbb{I} \left[\sum_{i \in \mathcal{C}} w_i \leq W^* \right] \times \sum_{i \in \mathcal{C}} v_i \quad (2.1)$$

and the objective is of the form $F(\mathcal{C}) > F^*$.

In general, for a given instance, one can ask the following questions:

Decision Does a configuration that realizes the objective exist?

Optimization What is the “tightest” objective which can be realized? For example, the largest value of F^* .

Search Which configuration realizes the objective?

Enumeration How many configurations realize the objective?

Approximation Which configuration realizes a weaker form of the objective, for example $F(\mathcal{C}) > \gamma F^*$ for some constant $\gamma < 1$?

The knapsack example above is a combination of an optimization problem (finding the largest possible value which can be realized) and a solution one (finding the corresponding configuration). Of course, one could ask many more questions. These are just the ones I shall be interested in in the following.

Let me cite a few more examples of problems:

Number Partitioning Given a set of N positive integers $S = \{n_i \in \mathbb{N}, i = 1, \dots, N\}$, find a subset $S' \subset S$ such that $\sum_{i \in S'} n_i = \sum_{i \in S \setminus S'} n_i$.

Subset Sum Given a positive integer K and a set of N positive integers $S = \{n_i \in \mathbb{N}, i = 1, \dots, N\}$, find a subset $S' \subset S$ such that $\sum_{i \in S'} n_i = K$.

Integer Linear Programming (ILP) Given a n -component real vector \mathbf{c} , a $n \times m$ real matrix \mathbf{A} , and a m -component real vector \mathbf{b} , find a n -component vector \mathbf{x} with non-negative integer components and which maximizes $\mathbf{c} \cdot \mathbf{x}$ subject to the constraints $\mathbf{Ax} \leq \mathbf{b}$.

Is Prime Given a positive integer N , determine if N is prime.

Many combinatorial optimization problems are defined on *graphs*. A graph \mathcal{G} is a double set of points, called *vertices*, $v \in \mathcal{V}$, and of distinct segments connecting pairs of points in \mathcal{V} , called *edges*, $e \in \mathcal{E}$: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Three special kinds of graphs are *cycles*, i.e. loops; *trees*, which are connected graphs that contain no cycles; and *bipartite* graphs, in which the set of vertices is divided in two, $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, and all edges have an endpoint in \mathcal{V}_1 and the other in \mathcal{V}_2 . Let me just mention a few important problems defined on graphs:

Hamiltonian Cycle (HC) Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, find a cycle $\mathcal{G}' \subset \mathcal{G}$ containing all the vertices of \mathcal{G} .

Traveling Salesman Problem (TSP) Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a weight $w(e) \in \mathbb{R}^+$ associated to each edge, find a HC with minimum total weight.

Minimum Spanning Tree (MST) Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a weight $w(e) \in \mathbb{R}^+$ associated to each edge, find a tree $\mathcal{G}' \subset \mathcal{G}$ containing all the vertices of \mathcal{G} with minimum total weight.

Vertex covering (VC) Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, find a subset $\mathcal{V}' \subset \mathcal{V}$ of the vertices of \mathcal{G} such that each edge $e \in \mathcal{E}$ has at least one of its endpoints in \mathcal{V}' , and minimizing $|\mathcal{V}'|$.

q -Coloring (q -COL) Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, assign to each vertex a color $c \in \{1, 2, 3, \dots, q\}$ such that no edge in \mathcal{V} has two endpoints of the same color.

Matching Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a weight $w(e) \in \mathbb{R}^+$ associated to each edge, find a subgraph $\mathcal{G}' \subset \mathcal{G}$ such that each vertex in \mathcal{V}' has one and only one edge in \mathcal{E}' , and which maximizes the total weight. Often \mathcal{G} is bipartite, in which case the problem is called *bipartite matching*.

Max Clique Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, find its largest *clique*, i.e. fully connected subgraph.

Min (or Max) Cut Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a weight $w(e) \in \mathbb{R}^+$ associated to each edge, find a partition $(\mathcal{V}_1, \mathcal{V}_2)$ of \mathcal{V} such that the total weight of the edges that have an edge in \mathcal{V}_1 and the other in \mathcal{V}_2 is minimized (or maximized).

All these problems are interesting from the theoretical point of view, and relevant for their practical applications. A further family of problems concerns *boolean satisfiability*, which I shall introduce in the next Section. The importance of boolean satisfiability problems and their connection to the other problems will be discussed in Section 2.3.

2.2 Boolean satisfiability: k -sat and k -xorsat

Boolean satisfiability problems are concerned with the following general question: given a boolean function $\mathcal{F}(x)$ over N boolean variables $x \equiv (x_1, \dots, x_N) \in \{\text{TRUE}, \text{FALSE}\}^N$, is there an assignment of the variables which makes the function evaluate to TRUE? The different problems of the family correspond to specific choices of the form of the function \mathcal{F} .

2.2.1 Introduction to k -sat

The prototype of satisfiability problems is the following. Given a N -tuple of boolean variables $x = (x_1, \dots, x_N)$, a *literal* is defined as a variable or its negation, e.g. x_3 and \bar{x}_7 ; a *k -clause* (or simply

clause, of length k) is defined as the disjunction of k literals, e.g. for $k = 3$: $x_2 \vee \bar{x}_4 \vee x_7$; finally, a *formula* is defined as the conjunction of M clauses. For example, for $N = 7, M = 3$:

$$\mathcal{F}(x) = (\bar{x}_3 \vee x_5 \vee \bar{x}_6) \wedge (x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee \bar{x}_5 \vee x_7). \quad (2.2)$$

Such a formula is said to be in *conjunctive normal form* (CNF), which is defined as

$$\mathcal{F}(x) = \bigwedge_{m=0}^M \left[\left(\bigvee_{j \in \mathcal{I}_m} x_j \right) \vee \left(\bigvee_{j' \in \mathcal{I}'_m} \bar{x}_{j'} \right) \right] \quad (2.3)$$

where \mathcal{I}_m and \mathcal{I}'_m are subsets of $\{1, \dots, N\}$ such that $\mathcal{I}_m \cap \mathcal{I}'_m = \emptyset$ for each $m = 1, \dots, M$.

The *satisfiability* problem (SAT) is the problem of determining if a given CNF formula admits at least one satisfactory assignment (also called a *solution*) or not. An interesting special case is that in which all the clauses have the same length k , in which case the problem is known as k -SAT. If the answer is “yes”, the formula is said to be *satisfiable*, which I shall denote by SAT¹, otherwise it is *unsatisfiable* which I shall denote UNSAT.

The same questions apply to k -SAT as to any other combinatorial optimization problem, namely the decision, optimization, solution, enumeration, and approximation problems, where the quantity to be minimized is the number of violated clauses.

A lot of attention has been devoted to k -SAT, principally for three reasons: first, for its theoretical relevance; many problems, from theorem proving procedures in propositional logic (the original motivation for k -SAT), to learning models in artificial intelligence, to inference and data analysis, can all be expressed as CNF formulæ. Second, because it is directly involved in a large number of practical problems, from VLSI circuits design to cryptography, from scheduling to communication protocols, all of which actually require solving or optimizing real instances of k -SAT formulæ. Third, and probably most notably, because of its central role in complexity theory, which I shall discuss in the next Section.

The questions of interest in the study of k -SAT can be divided in two broad families: on one hand those regarding the general properties of CNF formulæ and of their solutions (when they exist); on the other hand, those concerning the algorithms capable of answering the different questions one may ask (decision, optimization, ...); and of course, the intersection of the two (for example, proving that a certain algorithm succeeds in finding a solution under some assumptions also proves that a formula verifying those same assumptions must be SAT).

Also the answers that one can seek can be divided in two (or rather, their qualitative types): on one hand the results that are true in general and for any instance of k -SAT (under certain conditions), and on the other hand results that are true in a *probabilistic* way. Let me clarify this last case with an example. Suppose one considers the ensemble of all possible k -SAT formulæ with given N and M , with uniform weight. The total number \mathcal{N}_C of k -clauses that one can form with N variables is given by the number of choices of k among N indices times the number of choices for the k negations, i.e.

$$\mathcal{N}_C = \binom{N}{k} 2^k. \quad (2.4)$$

The number of formulæ \mathcal{N}_F that can be made with k independently chosen clauses is then

$$\mathcal{N}_F = (\mathcal{N}_C)^M. \quad (2.5)$$

Consider now a clause \mathcal{C} in the formula, for simplicity $\mathcal{C} = x_1 \vee \dots \vee x_k$. This clause will be satisfied by any of the 2^k possible values of (x_1, \dots, x_k) *except* the one corresponding to $x_i = \text{FALSE}$ for $i = 1, \dots, k$:

¹The use of SAT to designate both the general satisfiability problem and the satisfiable property of a formula should not lead to confusion, since in the future I shall be concerned exclusively with k -SAT.

out of all the possible assignments, only a fraction $1 - 1/2^k$ will satisfy any given clause. Since the formula contains $M \equiv \alpha N$ clauses (where α is defined as the ratio M/N), the *average* number of satisfying assignments will be

$$\mathcal{N}_S = 2^N \times \left(1 - \frac{1}{2^k}\right)^M = \left[2 \left(1 - \frac{1}{2^k}\right)\right]^{\alpha N}. \quad (2.6)$$

If we consider *large* formulæ, i.e. the limit $N \rightarrow \infty$, we see that the average number of solutions tends to 0 if

$$\alpha > -\frac{\log 2}{\log(1 - 2^{-k})}. \quad (2.7)$$

Notice that the average number of solutions is larger than or equal to the probability that a formula is SAT, since

$$\mathcal{N}_S = \sum_{n=0}^{2^N} n \times \mathbb{P}[\text{The number of solutions is } n] \geq \sum_{n=1}^{2^N} \mathbb{P}[\text{The number of solutions is } n] \quad (2.8)$$

and the sum on the right hand side is the probability that a formula is SAT. Therefore, we see that *in the limit $N \rightarrow \infty$ a random k -SAT formula chosen with uniform weight among all those with $M = \alpha N$ clauses is UNSAT with probability 1 if $\alpha > -\log 2 / \log(1 - 2^{-k})$.*

This kind of statement is very useful to characterize the *typical* properties of k -SAT formulæ under some given conditions. In many cases, the typical behavior is the interesting one, as it dominates the observable phenomena. The problem of studying k -SAT formulæ extracted from some distribution is often called Random- k -SAT. If the distribution is not specified, the uniform one is assumed.

Many interesting properties are easily proved for Random- k -SAT. For example, for $\alpha \rightarrow 0$ the probability $P_{\text{Sat}}(\alpha)$ that a random formula is SAT tends to 1. And it must be a decreasing function of α , since the property of being SAT is *monotone*: in order for a formula to be SAT, any sub-formula (made with a subset of its clauses) has to be satisfiable as well. In other words, adding clauses to a formula can only decrease its chances of being SAT, and adding random clauses to a random formula can only decrease its probability of being SAT.

From the physicist's point of view, probabilistic results are most interesting, because a random distribution of formulæ can be treated as a disordered system with some distribution of disorder. Indeed, one can represent Random- k -SAT as a spin glass. Each variable x_i will correspond to an Ising spin σ_i , which will be 1 if $x_i = \text{TRUE}$ and -1 otherwise. For a given configuration, the number of violated clauses will play the role of the energy:

$$E(\sigma) = \sum_{m=1}^M \prod_{j=1}^k \frac{1 + J_j^m \sigma_{i_j^m}}{2} \quad (2.9)$$

where i_j^m is the index of the j^{th} variable appearing in the m^{th} clause, and J_j^m is 1 if the variable appears negated and -1 otherwise. The set of $\{J_j^m\}$ and $\{i_j^m\}$ defines some random couplings which involve terms with $1, 2, \dots, k$ spins, have unit strength, and are attractive or repulsive with equal probability. As usual with statistical mechanics systems, we shall be interested in the thermodynamic limit $N \rightarrow \infty$. Since a random configuration violates a random clause with probability 2^{-k} , the energy is extensive (i.e. proportional to N) if α is of order $O(1)$ as $N \rightarrow \infty$. This is a perfectly legitimate *diluted* spin glass model. In fact, in Chapters 3 and 5 I shall present some results on Random- k -SAT obtained applying the replica method of Paragraph 1.4.3 to 2.9.

2.2.2 Introduction to k -xorsat

Another interesting boolean satisfiability problem goes under the name of k -XORSAT, and is obtained when the boolean function $\mathcal{F}(x)$ is the boolean equivalent of a linear system of equations:

$$\mathcal{F}(x) = \bigwedge_{m=1}^M \left[\left(\bigoplus_{j=1}^k x_{i_j^m} \right) \oplus y_m \right] \quad (2.10)$$

where the symbol \oplus denotes the logical operation XOR, and where $i_j^m \in \{1, \dots, N\}$ for $m = 1, \dots, M$ and $j = 1, \dots, k$ are some variable indices, and where $y \equiv (y_1, \dots, y_M)$ is some constant boolean vector. If we make the correspondence TRUE = 1 and FALSE = 0, this formula is equivalent to the linear system

$$\begin{cases} x_{i_1^1} \oplus \dots \oplus x_{i_k^1} = y_1, \\ x_{i_1^2} \oplus \dots \oplus x_{i_k^2} = y_2, \\ \dots \\ x_{i_1^M} \oplus \dots \oplus x_{i_k^M} = y_M. \end{cases} \quad (2.11)$$

An immediate consequence of this remark is that a very efficient algorithm is available to find if a given k -XORSAT formula is SAT, which assignments are solutions, and what is their number: the Gauss elimination procedure. One may even wonder why such a problem is interesting at all, given that it is equivalent to linear boolean algebra. The reasons are threefold: first, k -XORSAT is less easy than it seems. For example, if one determines with the Gauss elimination procedure that a k -XORSAT instance is not satisfiable, he could be interested in finding an approximate optimal configuration, i.e. an assignments which is guaranteed to satisfy a fraction $1 - \epsilon$ of the maximum possible number of clauses, for some given $\epsilon > 0$. Such an approximation algorithm, however, is not known (or rather, no such algorithm is known to work *efficiently*, the meaning of which will become clear in the next Section). Second, many questions regarding the *dynamics* of algorithms that can be applied to both k -SAT and k -XORSAT are interesting, difficult to answer for k -SAT, more manageable for k -XORSAT, and *a priori* should have at least qualitatively similar answers for the two problems. In these cases, k -XORSAT constitutes an excellent starting point to understand what happens in k -SAT. Finally, and foremost from the point of view of physicists, because k -XORSAT is a legitimate, and very interesting, spin glass model in its own. In fact, the diluted Ising p -spin model with couplings ± 1 is k -XORSAT: defining the energy as the number of violated clauses (as for k -SAT) and using the correspondence between boolean variables and Ising spins, we have

$$E(\sigma) = \sum_{m=1}^M \frac{1 - y_m \sigma_{i_1^m} \dots \sigma_{i_k^m}}{2}. \quad (2.12)$$

As in the case of k -SAT, the spin glass model is defined for some distribution of disorder, corresponding to an ensemble of possible k -XORSAT formulæ with a given measure, and we shall consider the thermodynamic limit $N \rightarrow \infty$ with some finite $\alpha = M/N$.

2.3 Computational complexity

Introducing k -XORSAT, I made the following implicit statement: that since an efficient algorithm for solving it was known, it could possibly be regarded as a less interesting problem than k -SAT. Is such a statement reasonable? Not really: whether a problem is “harder” than another or not should be an

intrinsic property of the problem, if it is meaningful at all, and should not be related to our knowledge (or lack thereof) of algorithms.

The question of what makes a problem intrinsically “hard”, and how to compare the “hardness” of different problems without introducing contingent dependencies (on the techniques and tools actually available to solve them) is the subject of computational complexity theory. It is a branch of rigorous mathematics, and it involves highly abstract (and quite complicated) models of computation. With no pretense in this direction, I shall only aim at giving the “flavor” of the most relevant concepts and results. An excellent (rigorous) introduction to the field is provided by the already cited reference [2].

2.3.1 Algorithms and computational resources

The first issue to be addressed is how to measure computational complexity. Let us consider that we have some decision problem, and an algorithm which can solve any instance of the problem. In order to compute the solution to the problem, the algorithm will use some *computational resources*. The most important of them is the *time* it will take to complete the computation. Other examples are the memory required to store the intermediate steps of the computation (usually referred to as *space*); some algorithms are probabilistic (we shall discuss them later), and require a supply of *random numbers*; in order to save space, some intermediate results may have to be erased, which has an *energy* cost (the loss of information corresponds to a decrease in entropy). There are several other relevant resources that one can consider. However, I shall consider only time.

In order to eliminate the dependency of the running time on such practical aspects as the hardware used to perform the computation or the actual code used to implement the algorithm, time will be defined as the number of elementary operations (such as arithmetic operations on single digit numbers, or comparisons between bits, *et cætera*) needed to complete the calculation. This will depend on the particular instance of the problem considered, and general results are obtained considering the *worst* possible instance for any given size n of the problem, and then taking the asymptotic behavior for large n . For example, if two different algorithms are available to solve the same problem, with times that scale as $t_1 \sim O(n^2)$ and $t_2 \sim O(n^3 \log n)$ respectively, then for large enough n it is sure that algorithm 1 will perform better than algorithm 2, regardless of the details of the dependency of t on n , and therefore of the specificities of the implementation.

Clearly, the main *theoretical* distinction will be between algorithm that have running times that increase as polynomials of the input size, and algorithms for which t increases as an exponential of the input size. This is easily seen by considering what happens to the “accessible” size of the input if the speed at which elementary operations are performed is increased by some constant factor, for different scaling behaviors of t versus n . This is done in Table 2.1. Notice, however, that *in practice* an algorithm running in time scaling as $10^3 n^3$ will take much longer than one scaling as $2^{10^{-3}n}$ for n up to $\simeq 10^4$. The point is that in the analysis of known algorithms, such “extreme” coefficients never occur.

2.3.2 Computation models and complexity classes

The analysis of algorithms provides (constructive) upper bounds on the computational resources required by the algorithm to solve some problem. A more interesting (and challenging) question would be to find some *lower* bound on the resources needed to perform some computation, independently on the algorithm used, which would then be a property of the problem itself. The theory of computational complexity tries to answer this question.

t	$n_a(1)$	$n_a(100)$	$n_a(10000)$
$O(n)$	n_1	$100 \times n_1$	$10000 \times n_1$
$O(n^2)$	n_2	$10 \times n_2$	$100 \times n_2$
$O(n^3)$	n_3	$4.6 \times n_3$	$21.5 \times n_3$
$O(2^n)$	n_4	$n_4 + 6.6$	$n_4 + 13.3$
$O(2^{2^n})$	n_5	$n_5 + 3.3$	$n_5 + 6.6$

Table 2.1: Increase of the “accessible” problem sizes for different scalings of running time, and for different increases in the computer speed. The first column reports the scaling of t as a function of n for different algorithms; the second column is the size of problems that can be computed in some given maximum time, which is denoted by n_i ; the third column reports the value of n_i obtained if the computer speed is increased by a factor 100; the last column corresponds to a factor of 10000. Notice that while polynomial algorithms have accessible sizes that increase by a constant *factor*, for exponential algorithms the increase is an *additive* constant.

In order to do that, *computation models* are introduced, which define what can (and cannot) be done in a computation. The most celebrated example of computation model is the *Turing machine* [23], which consists of the following: a *tape*, made of an unlimited number *squares*, each of which can contain a symbol s from some finite alphabet Σ ; a *head* which reads the tape and can perform some action a on it, such as “write s in this empty square”, “move right one square”, “erase this square”, “halt” *et caetera*; an internal *state* of the head, which is an element q_i of a finite set $\{q_1, \dots, q_r\}$; finally, a *computation rule*, which associates to any pair (s, q_i) a pair $(a, q_{i'})$, where s is the symbol on the square currently under the head and q_i its internal state, depending on which, a is an action performed by the head and $q_{i'}$ is the new internal state of the head.

The computation begins with some input written on the tape, and proceeds according to the computation rule, until the computation ends (i.e. the head halts). The result of the computation is what is written on the tape at the end. Different computation rules will compute different quantities, i.e. solve different problems.

Notice that *any* decision problem can be expressed in such a way that the instance is a string written in the alphabet Σ and the output is YES or NO, and therefore can be addressed by a suitable Turing machine. For example, a graph can be represented by a string over the alphabet $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (, -,)\}$ by specifying the number of vertices and then for each edge, the pair of vertices it connects, for example: $5(1-3)(1-5)(2-4)(2-5)(3-5)$. The decision problem is then equivalent to identifying which strings correspond to instances for which the answer is YES, that is to say whether the input string is or not an element of the subset of possible strings for which the answer is YES. Since subsets of possible strings are often called *languages*, decision problems are also referred to as languages, or as set recognition problems.

There are many variants of the Turing machine, such as *binary* machines, working on the alphabet 0,1; or *multi-tape* machines (which have a finite number of tapes and heads, and for which the computation rule specifies the joint action of all of them); or *universal* machines, for which the computation rule is provided as an input on the tape (which can always be done, since the rule can be represented as a string). For most of them, it can be proved that they are equivalent to a simple Turing machine, with an overhead on running time which is at most polynomial in the input size. Moreover, many other computation models, sometimes drastically different from the Turing machine, have been proved to be equivalent to it. It is a well established belief (but far from provable), going

under the name of *Church-Turing thesis*, that any computation which can physically be performed can be represented by a Turing machine.

Another very important variant is the *non-deterministic* Turing machine, which is a Turing machine with a computation rule which is not single valued: the machine is able to “split” (creating an identical copy of itself) and perform different actions on different tapes. One can either interpret this as a Turing machine with an infinite number of heads and tapes and which can transfer an infinite amount of information from one tape to another, or as “the most lucky” Turing machine, which at each split only executes one of the possible actions prescribed by the computation rule, and such that it leads to the “best” answer for the problem. Such a computation model is not feasible in practice, but we shall see that is very important from the theoretical point of view. In the following, by *polynomial time* I shall always mean on a deterministic Turing machine, unless differently specified.

Since the Turing machine is such a general paradigm for computations, it can be used to define *complexity classes*, i.e. classes of problems that have similar complexity. There are many different complexity classes that are relevant, but we shall focus on two of them:

Deterministic Polynomial Time (P) The class P is defined as the class of all decision problems that can be solved in polynomial time by a deterministic (i.e. “normal”) Turing machine.

Non-deterministic Polynomial Time (NP) The class NP is defined as the class of all decision problems that can be solved in polynomial time by a non-deterministic Turing machine.

Some comments are in order. First, notice that these class definitions do not refer to any specific algorithm: it is the fact that it is *possible* to solve them under certain conditions which matters, not that we are able to do it. Notably, no polynomial time algorithm is known for *any* NP problem, so the possibility to solve them in polynomial time on non-deterministic Turing machines is a mere definition.

However, and this is the second point, it is a very *meaningful* definition: for most problems, it is clear whether a problem is in P, in NP, or in none of the two. For example, for k -SAT an obvious algorithm is polynomial on a non-deterministic Turing machine: proceed in steps, and assign a variable at each step, splitting between the assignments TRUE and FALSE, then simplify the formula, and verify that there are no contradictions (i.e. clauses which cannot be satisfied); if this happens, halt the corresponding head; if some head achieves to assign all the variables, then it has found a satisfying assignment and the answer is SAT; on the contrary, if all the heads halt before they have assigned all the variables, there is no satisfying assignment and the answer is UNSAT. This procedure is obviously polynomial, so k -SAT is in NP. On the other hand, we have seen that the Gauss elimination procedure is polynomial (on a normal computer, and therefore on a Turing machine as well), and so k -XORSAT is in P.

Third, notice that any problem which is in P is also, *a fortiori*, in NP. In fact, the question of whether P and NP are *equal* (i.e. if there exist polynomial time algorithms to solve any NP problem) is one of the central open problems in complexity theory. It is strongly believed that the answer is no, but no proof (or disproof) of this is known.

Fourth, an equivalent, and more “practical” definition of NP is the following: NP is the class of all problems for which it is possible to issue a *certificate* in (deterministic) polynomial time. A certificate is the the answer YES or NO for a specific configuration, provided as input together with the instance of the problem. In other words, NP problems are such that a candidate solution can be verified in polynomial time. Again, it is obvious that k -SAT is in NP, and that any problem in P is also in NP. The equivalence of the two definitions is easy to verify: if a certificate to a problem can

be issued in polynomial time, a non-deterministic Turing machine can test in parallel all the possible configurations and find if some of them has answer YES. On the other hand, if a non-deterministic Turing machine can solve in polynomial time a problem, it can also check if any of the configurations for which the answer is YES coincides with the configuration submitted for the certificate.

Finally, notice that these definitions, given for decision problems, actually extend to search and optimization problems, so that if a decision problem belongs to NP (or P), then all of them are in the same class. For example, the optimization problem of k -SAT consists in finding the smallest value of E such that the decision problem “An assignment which satisfies $M - E$ clauses exists” gives answer YES. One can solve in (non-deterministic) polynomial time for $E = 0$, then for $E = 1$ and so on, and find in (non-deterministic) polynomial time the smallest E . However, the complexity classes of *enumeration* problems are often different.

2.3.3 Reductions, hardness and completeness

A *reduction* is a polynomial time algorithm which maps an instance of some decision problem into an instance of some other decision problem, such that the two instances always have the same answer. More formally, let us consider two decision problems A and B . Recall that A (and also B) can be viewed as the subset of the strings over the alphabet $\{0, 1\}$ which describe the instances of the problem that give answer YES. Then, we can write $x \in A$ to mean that the string x represents an instance of problem A for which the answer is YES, denote by $|x|$ the length of the string x , and define functions that associate a string to another string, i.e. $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ (the superscript $*$ denotes the set of all the possible strings in the alphabet). A formal definition of reduction is the following:

Reduction A decision problem A *reduces* to the decision problem B , denoted by $A \leq_p B$, if there exists a function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$, computable in polynomial time $p(|x|)$, such that $x \in A \Leftrightarrow f(x) \in B$.

Notice that since the function is computable in time bounded by $p(|x|)$, we must have

$$|f(x)| \leq p(|x|). \quad (2.13)$$

The concept of reduction is very powerful, since it permits to relate the complexity of different problems. In particular, one can define problems that are “at least as difficult” as *any* problem in some class:

Hardness A decision problem A is *C-hard* for some computational complexity class C if for any problem $B \in C$, $B \leq A$.

Completeness A decision problem A is *C-complete* for some computational complexity class C if $A \in C$ and for any problem $B \in C$, $B \leq A$.

Loosely speaking, C -complete problems are the most difficult problems to solve in class C , and if an efficient (i.e. polynomial) algorithm is found for a C -complete algorithm, it can solve efficiently any problem in C (for which a reduction is known).

The importance of k -SAT in complexity theory is due to the following

Cook-Levin Theorem 3-SAT is NP-complete [24, 25].

This was the first result on NP-completeness, introduced the concept, and proved that several other problems, to which SAT can be reduced, where also NP-complete.

The proof of the Cook-Levin theorem is surprisingly simple, and emphasizes the (conceptual) importance of the non-deterministic Turing machine: it is simply a mapping of the time evolution of the Turing machine into a SAT formula, in which the interpretation of boolean variables is “The cell i contains the symbol j at time k in the computation”, or “The head is over cell i at step k in the computation”, and “The head is in state q_i at the step k of the computation” (where i, j, k act as variable indices). The proof shows how to form a legitimate SAT formula for any given non-deterministic Turing machine, and then that any SAT formula can be reduced to 3-SAT.

k -SAT proves a very powerful tool for reductions, because of its generality and simple structure. The following problems are easily proven NP-complete, by reducing k -SAT to them: Integer Linear Programming, Hamiltonian Cycle, Traveling Salesman, Max Clique, Max Cut, Vertex Covering, 3-Coloring, The list is very, very long.

The fact that so many important problems are in NP, and that no efficient algorithms are known (and probably exist) to solve them, seems very discouraging in view of applications. However, this need not be the case, as I shall point out in the following Paragraph.

2.3.4 Other measures of complexity

The complexity classes P and NP are defined in terms of the asymptotic behavior of the running time for the *worst* possible instance of any size n . In many practical problems, one can be satisfied if some much less stringent requirements are met:

- If the *typical* running time over some distribution of instances is polynomial.
- If an approximate optimal solution can be found in polynomial time for any approximation factor ϵ .

Average-case complexity theory studies the first question; the theory of complexity of approximation studies the second.

Many average-case complexity results analyze the average time that some given algorithm takes to solve an instance of a problem, for a given distribution of instances. It is often the case that a NP problem is solved in polynomial time *on average* over some “natural” distribution of instances. For example, for problems defined on graphs, one can form the uniform distribution over all graphs containing n vertices and with some average connectivity. Then, one can prove that 3-COL can be solved in *linear* time on average. Often, however, all the algorithms known for some NP problem take exponential time on average. Alternatively, one can study the probability with which an algorithm finds an answer in polynomial time.

A crucial point in average-case complexity theory is the choice of the distribution. For example, the best known algorithm for Subset Sum take exponential time if the n numbers in the set are taken uniformly in the range $[1, 2^n]$. However, if this range is extended to $[1, 2^{n \log^2 n}]$, the average time for the best algorithm becomes polynomial. Even in cases when the dependency on the distribution is less dramatic, it remains a crucial point. For example, the reductions that map many NP problems on k -SAT introduce a very peculiar structure in the k -SAT formulæ they generate, so that even though the distribution of the instances of the original problem is a natural one, the distribution of k -SAT formulæ that are obtained is almost *never* a natural one. Thus, even though k -SAT can be solved efficiently on average in many cases under natural distributions, these results do *not* extend to the problems that can be reduced to k -SAT. On the other hand, even when it is possible to characterize the distribution of k -SAT formulæ generated by some reduction, it is usually either impossible to find

an algorithm that is efficient on average on them, or even to perform the analysis of the average case. This poses a severe limitation to the applicability of average-case complexity results.

On the other hand, complexity of approximation results are very interesting in view of applications. They are, however, usually more technical than the results I have discussed, and beyond the level of this introduction. I shall only cite the Probabilistically Checkable Proof (PCP) theorem and its consequences on the approximability of MAX-3-SAT [26], which is the optimization problem of 3-CNF formulæ.

One of the two equivalent definitions of the class NP requires that NP problems can be certified in polynomial time. The following definition extends the same concept:

Probabilistically Checkable Proof (PCP) Given two functions $r, q : \mathbb{N} \rightarrow \mathbb{N}$, a problem L belongs to the class $\text{PCP}(r, q)$ if there is a polynomial time probabilistic function (called *verifier*) $V : \{0, 1\}^* \rightarrow \{0, 1\}$ which, given as an input: a string x ; a string π (called *proof*); a sequence of $r(|x|)$ random bits; and which uses a substring of π , of size $q(|x|)$ and chosen at random, to compute $V^\pi(x)$, and is such that

$$\begin{cases} x \in L \Rightarrow \exists \pi : \mathbb{P}[V^\pi(x) = 1] = 1 ; \\ x \notin L \Rightarrow \forall \pi : \mathbb{P}[V^\pi(x) = 1] \leq \frac{1}{2} . \end{cases} \quad (2.14)$$

In this definition, the proof π is the analogous of the candidate configuration in NP: it is some string which is provided as an input, and which, if well chosen, can prove that $x \in L$ (i.e. that the answer to the instance represented by x of the decision problem L is YES). The verifier $V(x)$ is the analogous of the algorithm which issues the certificate, i.e. it gives, in polynomial time, an answer which is YES or NO and which is related to the answer to the instance represented by x . However, $V(x)$ is probabilistic, that is to say, it is a random variable. The source of the randomness is provided by the $r(|x|)$ random bits used to compute $V(x)$. For the problems in $\text{PCP}(r, q)$, the distribution of the values of $V(x)$ verifies the condition (2.14). Finally, notice that only a number $q(|x|)$ of symbols in π is actually used in the computation of $V^\pi(x)$, and these symbols are chosen at random.

At first sight, the class PCP seems very unnatural, and of little interest. The following theorem proves this impression very much wrong:

PCP Theorem $\text{NP} = \text{PCP}(O(\log n), O(1))$.

Again, several remarks. First, notice that any mathematical statement can be represented by a string, and that any mathematical proof can be represented by another string. Mathematical statements can be divided in two: right ones (i.e. theorems), and wrong ones. One can consider the following decision problem, called THEOREM: given a mathematical statement, is it a theorem? It is clear enough that it is possible to verify if a proof provided to support a statement is correct or not in a time which is polynomial in the length of the proof. Therefore, THEOREM is in NP.

What this theorem states is that any theorem represented by a string x can be recognized by looking at a *finite* number of randomly chosen bits of some suitable proof, represented by a string π , and evaluating some polynomial time function V . Then, if $V^\pi(x) = 0$ the statement is not a theorem with probability 1, while if $V^\pi(x) = 1$ it may or may not be. Conversely, if the statement x is a theorem, then there must be some proof π such that $V^\pi(x) = 1$ with probability 1, and if the statement is *not* a theorem, the probability that $V^\pi(x) = 1$ is less than or equal to $1/2$ for *any* proof string π . One can therefore check if the proof of any theorem (of any length) is correct just by looking at a *finite* number of bits in the proof, provided it is put in a suitable form, and obtain a probabilistic

result which is correct with probability 1 if the answer is NO, and correct with probability p if the answer is YES, for any $p < 1$.

Second, the same reasoning applies to *any* NP decision problem, not just THEOREM. For example, if, instead of providing a candidate solution to check if an instance of k -SAT is satisfiable, one provided a PCP proof π , then it would be possible to check it in *constant* time, rather than polynomial, obtaining a probabilistic result.

Third, even though the PCP theorem is very surprising in itself, the following corollary is also remarkable:

Hardness of approximation of MAX-3-SAT The PCP Theorem implies that there exists $\epsilon > 0$ such that $(1 - \epsilon)$ -approximation of MAX-3-SAT is NP-hard.

In other words, it is at least as difficult to find an approximation to the optimal assignment as it is to find the optimal assignment itself (if the approximation has to be good enough).

The theory of complexity of approximation is very rich and well established. However, I shall not discuss it any further.

2.3.5 Connections to the work presented in Part II

In Chapter 4, I shall present some results about what a certain class of algorithms can and cannot do on average for k -XORSAT, and also for an extension of k -XORSAT which is NP-complete.

The motivation for the work in presented in Chapter 5 is a recent result which establishes a relation between the *average-case* complexity for 3-SAT on the uniform distribution, and the *worst-case* complexity of approximation for several problems. The results I shall present provide an indication that some hypothesis, on which the previous relation is based, might be wrong.

2.4 Search algorithms

In the previous Section, I have introduced the concept of computational complexity, which measures how difficult it is to solve a problem. In this Section, I shall introduce several algorithms that attempt to do it in practice for the search problems associated to k -SAT and k -XORSAT, that is to say algorithms which try to find satisfying assignments for a given formula.

There is a huge variety of approaches and “strategies” to solve combinatorial optimization problems, and notably k -SAT. It is important to notice that, due to their formal similarity, the vast majority of the algorithms that can solve k -SAT can also solve k -XORSAT and *vice versa*, although with different performances (sometimes dramatically). I shall therefore discuss the two problems jointly, specifying the cases in which there are notable differences.

This introduction will be far from exhaustive: I shall focus on those algorithms of interest in view of the discussion of Part II. They can be divided in broadly two *families*:

Random-walks In random walks, all the variables are assigned at the first step of execution, typically at random, or following some more refined rule. In the following steps, single variables or groups of variables are selected and “flipped” (i.e. their value is changed), according to some stochastic rule which depends on the configuration. The algorithm stops when a solution is found, or when an upper bound to the number of steps has been reached. An algorithm in this family is specified by the rule according to which variables are flipped.

DPLL Procedure In the DPLL procedure, variables are assigned *sequentially*: at each step, a variable is selected according to some *heuristic* rule, and its value set according to some *strategy*. Once a variable is assigned, the formula is simplified by replacing it with its value. Under this process, the formula therefore evolves into a shorter and mixed one (i.e. including clauses of different lengths). Two events are especially important in the DPLL procedure: the generation of *unit clauses* and of *contradictions*. An algorithm in the DPLL family is specified by these four characteristics: the heuristic, the strategy, the action taken in the presence of unit clauses, and that in presence of contradictions.

The rest of this Section is organized in two Paragraphs, one for each family of algorithms. In each case, I shall consider the average case performance over the uniform distribution of instances, for either k -SAT or k -XORSAT.

2.4.1 Random-walk algorithms

The most familiar random-walk algorithm for physicists is the Metropolis Monte-Carlo procedure, which is capable of sampling configuration with probability equal to their Gibbs weight. In particular, the zero temperature version of the Metropolis algorithm consists in picking at each step a variable at random and flipping it if this decreases the number of violated clauses, and is a very simple example of “greedy” algorithm, i.e. an algorithm which tries to perform a local optimization of the configuration at every move.

Based on the qualitative arguments about frustration presented in Paragraph 1.2.4, such a *local optimization* procedure is bound to fail in disordered systems. The following arguments shows that this is the case with probability 1 for uniformly drawn random instances of 3-XORSAT. Consider the subformula represented in Figure 2.1, which I shall call a “blocked island”. It is clear that if such a subformula is present in the formula, and if it is found in a configuration such as one of those depicted in the figure, a greedy algorithm will not be able to reach a satisfying assignment. In [27] it is shown that in the limit $N \rightarrow \infty$ this situation occurs with finite probability

$$p = \frac{729}{1024} \alpha^7 e^{-45\alpha} \quad (2.15)$$

where α is the clause to variable ratio, $\alpha \equiv M/N$. The average number of blocked islands in a random 3-XORSAT formula is $pN = O(N)$, and it is a lower bound to the minimum number of violated clauses of configurations that greedy algorithms are able to find.

More interesting are “less greedy” algorithms. A simple example is provided by Pure Random Walk Sat (PRWalkSAT), which was introduced in [28], and is defined as follows: initially, assign all the variables uniformly at random; then, at each step pick uniformly at random a clause among those that are violated, and a variable among those appearing in it, and flip it; repeat, until a satisfying assignment is found, or a number T_{\max} of steps has been performed. Notice that by flipping a variable which appears in a violated clause, that clause becomes satisfied; however, if that variable also appear in other clauses that were satisfied before the flip, they might become unsatisfied after. This is why this algorithm is “less” greedy. The possible outcomes of the algorithm are two: either a satisfying assignment is produced, or the output is UNDETERMINED.

In [28], it was shown that PRWalkSAT finds a solution with probability 1 for *any* satisfiable instance of 2-SAT in a number of steps (i.e. time) of order $O(N^2)$. An interesting extension of this result to 3-SAT was obtained in [29], where it is shown that if $T_{\max} = 3N$ and the procedure is repeated for a number R of times without obtaining a satisfying assignment, then the probability that the instance is

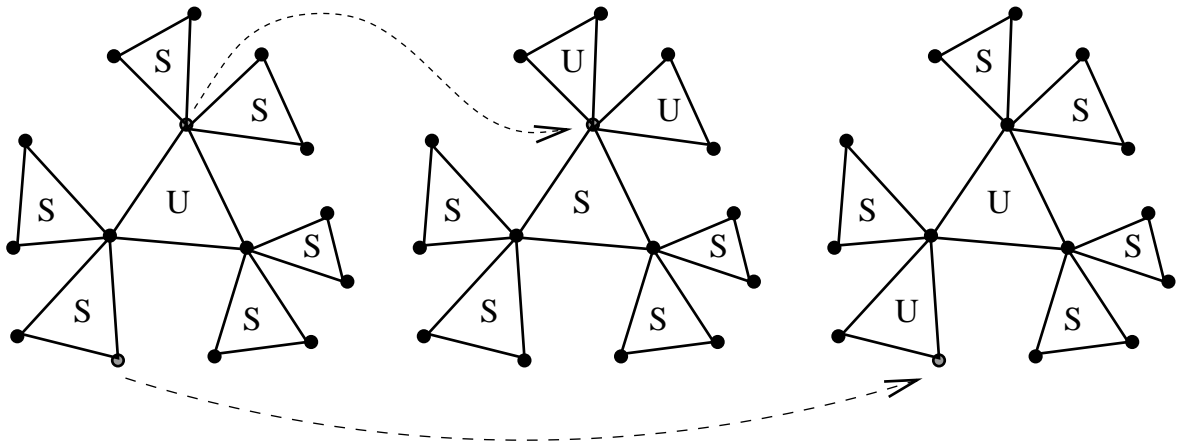


Figure 2.1: Representation of a “blocked island”. Each dot in the diagram corresponds to a variable, and triangles represent 3-clauses containing the variables at the vertices. The left-most diagram shows an isolated subformula; the variables in the subformula are all assigned, in such a way that the clauses marked with the letter S are satisfied, those with the letter U are unsatisfied. If one of the variables appearing in the central clause is flipped, the second configuration is obtained; if one of the variables which do not appear in the central clause is flipped, the third configuration is obtained. In both cases, the number of unsatisfied clauses increases by 1 (From [27]).

SAT is upper-bounded by $\exp[-R(3/4)^N]$. By taking R sufficiently larger than $(4/3)^N$, the probability that an instance for which no satisfying assignment has been found is nonetheless satisfiable can be made arbitrarily small. Also, notice that, even though the running time of such a procedure (for any fixed probability bound) is exponential, it is still exponentially smaller than 2^N , which would be required by exhaustive search.

The previous results hold for *any* instance, and the probabilities mentioned are over the choices of the algorithm. Another interesting question is to analyze the average-case behavior over the uniform distribution of k -SAT instances. This was done in [30, 31, 32]. In the first of these papers, a rigorous bound is found for the values of the clause-to-variable ratio $\alpha \equiv M/N$ for which PRWalkSAT finds a solution in polynomial time with probability 1: $\alpha < \alpha_{\text{PRWalkSAT}} \simeq 1.63$ (for $k = 3$). This is the first example I mention of an algorithmic bound on α , i.e. a threshold value separating two different behaviors of the same algorithm. Many more will follow. Also, notice that since with probability 1 PRWalkSAT finds a solution for random 3-SAT formulae with $\alpha < \alpha_{\text{PRWalkSAT}}$, this implies that these formulae are satisfiable with probability 1.

In [31, 32] the same problem was studied with “physical” methods. In particular, a numerical study indicates that random instances are solvable with probability 1 in polynomial time if $\alpha \lesssim 2.7$, while for larger values the time becomes exponential. The analysis of the master equation performed in [31] shows that the average fraction of unsatisfied clauses, $\varphi(t)$, after tN steps of the algorithm, is a deterministic function which depends on α and goes to 0 in finite t if $\alpha \lesssim 2.7$, while for larger α it tends asymptotically to a finite value, which is 0 for $\alpha \simeq 2.7$ and then increases. In this second regime, it can happen that solutions are found because of *fluctuations*, but the time which this requires is exponential in N .

A somewhat more complicated variant of this algorithm goes under the name of WalkSAT, and is defined as follows:

```

procedure WalkSAT( $p, T_{\max}$ )
  Assign uniformly at random each variable
  repeat
    Select uniformly at random a clause  $\mathcal{C}$  which is UNSAT
    For each variable  $x_i$  in  $\mathcal{C}$ , compute the break-count  $b(x_i)$ , defined as the number of clauses
    currently satisfied that will be violated if  $x_i$  is flipped
    if A variable  $x_j$  in  $\mathcal{C}$  has break-count  $b(x_j) = 0$  then
      Flip  $x_j$ 
    else With probability  $p$  :
      Select the variable in  $\mathcal{C}$  with the lowest break-count (or select uniformly at random one
      of the variables with the lowest break-count, if there are more than one), and flip it
    else With probability  $1 - p$  :
      Select uniformly at random a variable in  $\mathcal{C}$  and flip it
    end if
  until There are no UNSAT clauses, or the number of steps exceeds  $T_{\max}$ 
  if A solution  $X$  has been found then return  $X$ 
  else return UNDECIDED
  end if
end procedure

```

As in the case of PRWalkSAT, variables to be flipped are selected only in clauses that are currently UNSAT. However, instead of picking a variable at random, WalkSAT looks for a variable which can be flipped *without making any clause UNSAT which is currently SAT*. Notice that in doing this the total number of UNSAT clauses must decrease of at least 1 (i.e. the selected clause becoming SAT). On the other hand, if some clauses currently SAT have to become UNSAT, the variable which minimizes their number is selected, with probability p , or otherwise any variable in the clause uniformly at random. In both of these cases, the total number of UNSAT clauses can increase.

The average case performance of WalkSAT is astonishingly good. Numerical experiments suggest that its typical running time (e.g. the median over a series of runs) remains linear for α up to 4.15 (for $k = 3$) [33]. Interestingly, this value coincides with the threshold for the stability of the 1RSB solution [34].

For larger values of α , the behavior of WalkSAT becomes more complicated. The average running time becomes exponential, with a peculiar structure in the average fraction of unsatisfied clauses as a function of the number of steps (divided by N). A detailed analysis of this behavior is the object of current work in collaboration with Giorgio Parisi.

2.4.2 DPLL algorithms

The DPLL procedure is a firmly established complete algorithm for k -SAT and similar constraint satisfaction problems. For concreteness, and for future reference in Chapter 4, I shall consider the case of k -XORSAT. DPLL was introduced by Davis and Putnam in 1960 [35] and developed by Davis, Logemann and Loveland in 1962 [36], and has many variants.

The basic principle is to assign the variables in sequential order, and simplify the formula after each assignment. This generates a sub-formula in which clauses that are satisfied are eliminated, and clauses in which the assigned variable appears decrease in length of one unit. If a unit clause is generated (i.e. a clause of length 1), this clause determines the value of the variable appearing in it, and it is assigned accordingly. This event is called *Unit Propagation* (UP). The rule according to which

the variable to be assigned is selected is called *heuristic*. Most often, the value assigned is selected uniformly at random, but sometimes a rule, called *strategy*, determines it. The simplest example of heuristic consists in selecting the variable uniformly at random among those not yet assigned, as well as the value, but giving priority to UP; it is called Unit Clause (UC).

A crucial distinction between DPLL variants is the action taken if a *contradiction* arises, i.e. in the case of k -XORSAT, a pair of unit clauses for the same variable with conflicting assignments. If this occurs, no value of the variable in question will satisfy the subformula, and therefore the original one. This event signals that some of the previous assignments were wrong. Two possible actions can then be taken: either modify some of the previous assignments, or output UNDETERMINED and possibly restart the procedure. In the first case, the algorithm *backtracks* to the last variable which was set by a “free” step (as opposed to a UP or a backtrack), and inverts it. In the second case, the algorithm is no longer complete, but we shall see that it can still be interesting in the average case.

Formally, we can describe the DPLL procedure with and without backtracking with the two following procedures, in which \mathcal{F} is the formula and H is the heuristic, i.e. a function which associates an index of a variable not yet assigned to a subformula. With no backtracking,

```

procedure DPLL WITHOUT BACKTRACKING( $\mathcal{F}, H$ )
  repeat
    for every unit clause  $\mathcal{U}$  in  $\mathcal{F}$  do
       $\mathcal{F} \leftarrow \text{Simplify}(\mathcal{F}, \mathcal{U})$ 
     $i \leftarrow H(\mathcal{F})$ 
     $\mathcal{F} \leftarrow \text{Simplify}[\mathcal{F}, x_i = S(\mathcal{F})]$ 
    if a contradiction is present then
      return UNDETERMINED
  until all the variables are assigned
  return TRUE
end procedure

```

where $S(\mathcal{F})$ is the strategy according to which values for assignments are decided. With backtracking the procedure is somewhat more complicated, and it is more conveniently expressed in a *recursive* form:

```

procedure DPLL WITH BACKTRACKING( $\mathcal{F}, H$ )
  if all the the clauses are satisfied then
    return TRUE
  if a contradiction is present then
    return FALSE
  for every unit clause  $\mathcal{U}$  in  $\mathcal{F}$  do
     $\mathcal{F} \leftarrow \text{Simplify}(\mathcal{F}, \mathcal{U})$ 
   $i \leftarrow H(\mathcal{F})$ 
  return DPLL[Simplify( $\mathcal{F}, x_i = \text{TRUE}$ ),  $H$ ]  $\vee$  DPLL[Simplify( $\mathcal{F}, x_i = \text{FALSE}$ ),  $H$ ]
end procedure

```

The complete variant of DPLL (i.e. the one with backtracking) has been extensively studied (see for example [37, 38] and references therein). In the following, I shall concentrate on DPLL without backtracking.

Many different heuristics for DPLL have been studied, in view of both theoretical studies and applications. In the following, an important role will be played by the Generalized Unit Clause (GUC), introduced and studied in [39, 40, 41], which is defined as follows: at each step, select uniformly at

random a clause among those of *shortest length*, and then uniformly at random a variable in it. This generalizes the UP rule to clauses of length larger than unit, hence the name.

The analysis of the average case behavior of DPLL heuristics can be simplified considerably using the following approach, introduced in [42]. Consider the state of the formula after T variables have been set. It will contain a number $C_j(T)$ of clauses of length $j = 1, 2, \dots, k$ (for some values of T , some unit clauses will not have been removed yet, hence the term $j = 1$). The formula can be described as a table in which each row represents a clause, and each “slot” in it represents a variable. Initially, there are M rows, each of length k , which then become shorter as the algorithm proceeds. If the heuristics we consider consist in the selection of either a variable uniformly at random, or of a slot in the table according to some rule which does not depend on the content of the slots, then the subformulae that are generated are uniformly random conditioned on their lengths. This is the case of both UC (which always selects the variable uniformly at random) and of GUC (which selects, uniformly at random, first a row in the table among those of shortest length, and then a slot in the row).

In the case of UC, at each step a variable is selected uniformly at random. Because of the statistical independence of the subformulae, each slot has a probability $1/(N - T)$ of containing the selected variable, and a clause of length j has probability $j/(N - T)$ of containing it. Since the clauses of length j that contain the selected variable become of length $j - 1$, the *average* variation in the number of clauses is

$$\mathbb{E}[C_j(T + 1) - C_j(T) | \{C_j(T)\}] = \frac{(j + 1)C_{j+1}(T) - jC_j(T)}{N - T} \quad (2.16)$$

where, for notational simplicity, we set $C_{k+1}(T) \equiv 0$. Notice that this is the same equation one obtains for steps in which UP is applied, when instead of selecting the variable uniformly at random it is selected among those appearing in unit clauses.

A theorem by Wormald [43], the statement of which is rather technical and I shall omit, ensures that (under some very general assumptions which are satisfied by all the heuristics we shall consider) the clause *densities* are concentrated in the thermodynamic limit,

$$\mathbb{E}[C_j(T)] = Nc_j(T/N) \quad (2.17)$$

where $c_j(t)$ is a function determined by the differential equation obtained dividing (2.16) by $\Delta T = 1$.

$$\frac{dc_j(t)}{dt} = \lim_{N \rightarrow \infty} \frac{\mathbb{E}[\Delta C_j(T) | \{C_j(T)\}]}{\Delta T} = \frac{(j + 1)c_{j+1}(t) - jc_j(t)}{1 - t} \quad (j = 2, \dots, k). \quad (2.18)$$

Since the initial formula contains $M = \alpha N$ clauses of length k , the initial condition for this system of equations is $c_j(0) = \delta_{j,k}\alpha$. Notice that, if at any time, $c_1(t) > 0$, i.e. the formula contains an extensive number of unit clauses, each of them has a probability of order $1/N$ of containing any given variable, so that there is a finite probability that two unit clauses will contain the same variable. If this happens, a contradiction is generated with finite probability at each step of the algorithm, so that over a finite interval of time Δt this will happen with probability 1. Therefore, if at any time during the evolution of the formula $c_1(t)$ becomes positive, the algorithm will generate a contradiction and will stop. This is the reason why the range of values of j starts with 2. Since the rate at which unit clauses are generated is

$$\frac{2c_2(t)}{1 - t} \quad (2.19)$$

and the rate at which they are removed is at most 1 (because one variable is set at each time step, and therefore at most one unit clause is removed), the condition for the onset of contradictions is

$$\frac{2c_2(t)}{1 - t} = 1. \quad (2.20)$$

The system of equations (2.18) is easily solved:

$$c_j(t) = \alpha \binom{k}{j} (1-t)^j t^{k-j} \quad (j = 2, \dots, k). \quad (2.21)$$

The algorithm will provide a solution with probability 1 if all the variables are set without generating contradictions, i.e. if $2c_2(t)/(1-t) < 1$ for all $t \in [0, 1]$. For $c_2(t)$ given by (2.21), this function reaches a maximum for $t = t^* \equiv (k-2)/(k-1)$, in which its value is

$$\max_{t \in [0,1]} \frac{2c_2(t)}{1-t} = \alpha k \left(\frac{k-2}{k-1} \right)^{k-2} \quad (2.22)$$

which is equal to 1 if

$$\alpha = \alpha_h^{\text{UC}} \equiv \frac{1}{k} \left(\frac{k-1}{k-2} \right)^{k-2}. \quad (2.23)$$

Notice that this implies that for $\alpha \leq \alpha_h^{\text{UC}}$, random k -XORSAT formulæ from the uniform distribution are satisfiable with probability 1, and UC is capable in finding a satisfactory assignment in linear time with probability 1.

A similar analysis can be performed for GUC. Initially, the formula contains M clauses of length k . As variables are set, some clauses become shorter: let us suppose that after T steps the number of clauses of length j is $C_j(T)$ for $j = j^*, \dots, k$ with $j^* > 1$, and 0 for $j < j^*$, and let us consider what happens starting from there. When a variable is set by GUC, it is selected among the shortest clauses, i.e. those of length j^* . A clause of length $j^* - 1$ is generated, and the other numbers of clauses vary only if the same variable appears in other equations. That is to say, after the first variable has been set the average variations in $C_j(T)$ are:

$$\begin{aligned} \langle \Delta^{(1)} C_j(T) \rangle &\equiv \mathbb{E}[C_j(T+1) - C_j(T) | \{C_j(T)\}] \\ &= \frac{(j+1)C_{j+1}(T) - jC_j(T)}{N-T} \quad (j = j^* + 1, \dots, k), \end{aligned} \quad (2.24)$$

$$\begin{aligned} \langle \Delta^{(1)} C_{j^*}(T) \rangle &\equiv \mathbb{E}[C_{j^*}(T+1) - C_{j^*}(T) | \{C_j(T)\}] \\ &= -1 + \frac{(j^*+1)C_{j^*+1}(T) - j^*C_{j^*}(T)}{N-T}, \end{aligned} \quad (2.25)$$

$$\begin{aligned} \langle \Delta^{(1)} C_{j^*-1}(T) \rangle &\equiv \mathbb{E}[C_{j^*-1}(T+1) | \{C_j(T)\}] \\ &= 1 + \frac{j^*C_{j^*}(T)}{N-T}. \end{aligned} \quad (2.26)$$

where the superscript (n) indicates that n variables have been set (here, $n = 1$).

Notice that the average number of clauses of length $j^* - 1$ is now of order $O(1)$, and not smaller than 1. GUC will then select a clause from one of the clauses of length $j^* - 1$, giving:

$$\langle \Delta^{(2)} C_j(T) \rangle = 2 \frac{(j+1)C_{j+1}(T) - jC_j(T)}{N-T} + O(N^{-1}) \quad (j = j^* + 1, \dots, k), \quad (2.27)$$

$$\langle \Delta^{(2)} C_{j^*}(T) \rangle = -1 + 2 \frac{(j^*+1)C_{j^*+1}(T) - j^*C_{j^*}(T)}{N-T} + O(N^{-1}), \quad (2.28)$$

$$\langle \Delta^{(2)} C_{j^*-1}(T) \rangle = 2 \frac{j^*C_{j^*}(T)}{N-T} + O(N^{-1}), \quad (2.29)$$

$$\langle \Delta^{(2)} C_{j^*-2}(T) \rangle = 1 + O(N^{-1}). \quad (2.30)$$

In this equations, the terms $O(N^{-1})$ come from the fact that we are considering the initial T for evaluating the functions, which results in a variation of $O(1)$ in the values of the C_j . Notice that UP

do not contribute to values of j that are smaller than $j^* - 1$, because the number of clauses of such lengths are not extensive.

It will then take (on average) $j^* - 1$ steps (after the first one) to “empty” one of the clauses of length $j^* - 1$ that have been generated:

$$\langle \Delta^{(j^*)} C_j(T) \rangle = j^* \frac{(j+1)C_{j+1}(T) - jC_j(T)}{N-T} + O(N^{-1}) \quad (j = j^* + 1, \dots, k), \quad (2.31)$$

$$\langle \Delta^{(j^*)} C_{j^*}(T) \rangle = -1 + j^* \frac{(j^*+1)C_{j^*+1}(T) - j^*C_{j^*}(T)}{N-T} + O(N^{-1}), \quad (2.32)$$

$$\langle \Delta^{(j^*)} C_{j^*-1}(T) \rangle = j^* \frac{j^*C_{j^*}(T)}{N-T} + O(N^{-1}), \quad (2.33)$$

$$\langle \Delta^{(j^*)} C_{j^*-2}(T) \rangle = O(N^{-1}). \quad (2.34)$$

Let us call a *round* the sequence of steps starting with the assignment of a variable in a clause of length $j^* - 1$ and ending when there are no more clauses shorter than $j^* - 1$, such as the steps from 2 to j^* in the previous argument. Each round has the same duration: $j^* - 1$ steps. During such a round, the variation of the average number of clauses of length $j^* - 1$ is

$$\langle \Delta^{(\text{round})} C_{j^*-1}(T) \rangle = -1 + (j^* - 1) \frac{j^*C_{j^*}(T)}{N-T} + O(N^{-1}), \quad (2.35)$$

so that after $r \geq 1$ rounds the average variations will be

$$\langle \Delta^{(1+r(j^*-1))} C_j(T) \rangle = [1 + r(j^* - 1)] \frac{(j+1)C_{j+1}(T) - jC_j(T)}{N-T} + O(N^{-1}) \quad (j = j^* + 1, \dots, k), \quad (2.36)$$

$$\langle \Delta^{(1+r(j^*-1))} C_{j^*}(T) \rangle = -1 + [1 + r(j^* - 1)] \frac{(j^*+1)C_{j^*+1}(T) - j^*C_{j^*}(T)}{N-T} + O(N^{-1}), \quad (2.37)$$

$$\langle \Delta^{(1+r(j^*-1))} C_{j^*-1}(T) \rangle = 1 + \frac{j^*C_{j^*}(T)}{N-T} + r \left[-1 + (j^* - 1) \frac{j^*C_{j^*}(T)}{N-T} \right] + O(N^{-1}). \quad (2.38)$$

There are two possible cases: either after a finite average number R of rounds the average number of clauses of length $j^* - 1$ returns to 0, or not. In the first case, R is obtained from the condition:

$$\langle \Delta^{(1+R(j^*-1))} C_{j^*-1}(T) \rangle = 0 \quad (2.39)$$

$$\Leftrightarrow R = \frac{1 + \frac{j^*C_{j^*}(T)}{N-T}}{1 - (j^* - 1) \frac{j^*C_{j^*}(T)}{N-T}} + O(N^{-1}). \quad (2.40)$$

Notice that, since R is an *average* number, it needs not be integer, and also that the condition for R to be finite is

$$\frac{j^*C_{j^*}(T)}{N-T} < \frac{1}{j^* - 1}. \quad (2.41)$$

After R rounds, the number of steps that have been taken is

$$\Delta T = 1 + R \times (j^* - 1) = \frac{j^*}{1 - (j^* - 1) \frac{j^*C_{j^*}(T)}{N-T}} + O(N^{-1}) \quad (2.42)$$

and the total average variations will be:

$$\langle \Delta^{(\Delta T)} C_j(T) \rangle = \Delta T \frac{(j+1)C_{j+1}(T) - jC_j(T)}{N-T} + O(N^{-1}) \quad (j = j^* + 1, \dots, k), \quad (2.43)$$

$$\langle \Delta^{(\Delta T)} C_{j^*}(T) \rangle = -1 + \Delta T \frac{(j^*+1)C_{j^*+1}(T) - j^*C_{j^*}(T)}{N-T} + O(N^{-1}), \quad (2.44)$$

$$\langle \Delta^{(1+r(j^*-1))} C_{j^*-1}(T) \rangle = O(N^{-1}). \quad (2.45)$$

Wormald's theorem can be applied, ensuring that in the thermodynamic limit the contributions of order $O(N^{-1})$ are influential, and that the average densities are concentrated around the functions $c_j(t)$ that are solutions of the differential equations obtained by dividing (2.45) by ΔT , given by (2.42). The equations we obtain are the following:

$$\frac{dc_j}{dt} = \frac{(j+1)c_{j+1} - jc_j}{1-t} \quad (j = j^* + 1, \dots, k), \quad (2.46)$$

$$\frac{dc_{j^*}}{dt} = \frac{(j^*+1)c_{j^*+1} - j^*c_{j^*}}{1-t} - \frac{1}{j^*} \left[1 - (j^* - 1) \frac{j^*c_{j^*}}{1-t} \right] \quad (2.47)$$

which we can rewrite as a single equation

$$\frac{dc_j}{dt} = \frac{(j+1)c_{j+1} - jc_j}{1-t} + \delta_{j,j^*} \left[\frac{1}{j^*} - \frac{(j^* - 1)c_{j^*}}{1-t} \right] \quad (j = j^*, \dots, k). \quad (2.48)$$

We still have to analyze what happens when R diverges. In that case, the rate at which clauses of length $j^* - 1$ accumulate is larger than the rate at which they can be removed, and their number becomes extensive. This signals that the value of j^* must decrease by one unit.

In Paragraph 4.3.3 I shall give a detailed study of the solution to these equations for $k = 3$, showing that GUC finds solutions in linear time with probability 1 for random formulæ from the uniform distribution for $\alpha \leq \alpha_h^{\text{GUC}}(3) \simeq 0.750874$, which is therefore a lower bound for the value up to which random formulæ are SAT with probability 1.

Chapter 3

Phase transitions in random optimization problems

In the previous Chapter I have introduced two random optimization problems, k -SAT and k -XORSAT, which are equivalent to some spin glass models. In this chapter I am going to review the rich phenomenology displayed by these models, consisting of several phase transitions regarding different order parameters.

I shall first make a very brief introduction to the discovery of sharp transitions in numerical experiments, mostly concerning k -SAT, in Section 3.1; then, I shall give a rigorous derivation the phase diagram of k -XORSAT in Section 3.2; finally, in Section 3.3 I shall sketch the main results on the phase diagram of k -SAT.

3.1 Evidence of phase transitions from numerical experiments

Phase transitions are a common and well understood concept in statistical mechanics. In the context of random combinatorial optimization problems, it is far less obvious what this can mean. I shall therefore start with a definition and a simple example.

Let us consider a random problem defined over some distribution of instances, and a property \mathcal{P} which might be true or false for each instance. I shall denote by N the size of the problem, by c some control parameter, and by $P(N, c)$ the probability over the distribution of instances that \mathcal{P} is true. Then, a *sharp transition* in \mathcal{P} is defined by the following condition:

$$\lim_{N \rightarrow \infty} P(N, c) = \begin{cases} 0 & \text{if } c < c^* \\ 1 & \text{if } c > c^* \end{cases} \quad (3.1)$$

where c^* is a constant *threshold* independent on N .

For example, we might consider random graphs with N vertices and $M = cN$ edges, and ask what is the probability $P(N, c)$ that the largest connected component in the graph has size γN with $\gamma > 0$ and independent on N . This problem, called random graph percolation, has been studied by Erdős and Rényi in [49, 50]. They have proved that the percolation indeed undergoes a sharp transition, with threshold value $c^* = 1/2$.

In numerical studies the definition (3.1) is of little use, as the size of samples has to be finite. Some method to extrapolate results to the $N \rightarrow \infty$ limit is needed. For large but finite N , $P(N, c)$ will be

a smooth function of c varying from 0 to 1, whose form will in general depend on N . The *transition region*, defined as the range of values of c in which $\epsilon < P(N, c) < 1 - \epsilon$ for some finite ϵ independent on N , will have a width $\Delta(N)$ which will become smaller and smaller as N grows. If $\Delta(N)$ scales as a power of N , $\Delta(N) \sim N^{-\nu}$ for some constant ν , one can rescale

$$P(N, c) = \phi_N((c - c^*)N^\nu) \quad (3.2)$$

and hope that the function $\phi_N(\cdot)$ becomes independent of N for large (but experimentally accessible) N . If this is the case, the values of ν and c^* can be obtained by fitting numerical data so that they “collapse” on $\phi(c)$. This is one of the simplest applications of a general method which goes under the name of *finite size scaling* (see for example [51]). In the case of percolation on random graphs, a finite size scaling of the type of (3.2) holds, with $\nu = 1/3$.

Finite size scaling was applied in [52] to k -SAT, providing the first numerical evidence for a sharp transition between a SAT phase where random formulæ are satisfiable with probability 1 and a UNSAT phase where they are not satisfiable with probability 1. The threshold value $\alpha_s(k)$ was measured for $k = 2, 3, 4, 5, 6$, together with the exponent $\nu(k)$. For example, for $k = 3$ the values found were $\alpha_s(3) \simeq 4.17$ and $\nu(3) \simeq 0.67$. However, due to the relatively small size of the formulæ considered ($N \approx 100$), these values were later proved to be inaccurate (most notably the exponents).

Previous studies, for example [53], had measured the probability of a random formula being satisfiable, pointing out that it was $1/2$ for $\alpha \simeq 4.25$ for $k = 3$ and N sufficiently large, but without discussing the N dependence of the transition width. In fact, the main purpose of that study was to analyze a different phenomenon: the variation of the running times of the complete DPLL procedure on random formulæ as a function of α . What the authors had noticed, and motivated their work, was that formulæ were “hardest” to solve in a region centered on the value of α corresponding to $\mathbb{P}[\text{Sat}|N, \alpha] = 1/2$.

This problem was analyzed again in [54], in which finite size scaling techniques were applied to the median running time as a function of N and α . Even though the maximum of the running time is reached for $\alpha \simeq \alpha_s(k)$ for large N , this is a very different phenomenon from the SAT/UNSAT transition, since it is related to the dynamical properties of an algorithm (while the SAT/UNSAT transition is a property of the ensemble of formulæ themselves).

These two problems, the phase transitions of random constraint satisfaction problems, and the dependency on α of the performance of algorithms, as well as their connection to the properties of typical random formulæ and of their solutions, will be the main topic of the rest of this Chapter, in which I shall present some well known results, and of the second Part of this thesis, presenting some original ones.

3.2 Rigorous derivation of the phase diagram of k -XORSAT

In this section I shall present some rigorous results on the phase diagram of k -XORSAT. The cases $k = 1$ and $k = 2$ are much simpler than the general case $k \geq 3$. On the other hand, all values of $k \geq 3$ give rise to the same behavior (at least qualitatively), while the behavior for $k = 1$ and 2 is different. For these reasons I shall restrict $k \geq 3$ in this Chapter.

As in the case of k -SAT, it is intuitive to expect that as the ratio $\alpha = M/N$ between the number of clauses M and the number of variables N increases, the probability that a random formula be satisfiable will decrease. And numerical experiments confirm that (as was the case for k -SAT) the transition between the SAT and the UNSAT phases becomes sharp as $N \rightarrow \infty$.

However we shall see that the phase diagram of k -XORSAT presents a richer structure than just a SAT/UNSAT transition, and that the geometrical properties of the set of solutions in the SAT phase present a second phase transition, which can be related to the performance of search algorithm, as I shall discuss in Chapter 4.

3.2.1 Bounds from first and second moments

In this paragraph I shall derive a rigorous bound for the threshold value $\alpha_s(k)$ of the SAT/UNSAT transition, first proved in [55].

The number of solutions \mathcal{N} of k -XORSAT formulæ with fixed M and N can be regarded as a random variable whose distribution $\mathcal{P}(\mathcal{N})$ will depend on the distribution of the formulæ considered. Since this random variable only takes integer values, the following identity must hold:

$$\langle \mathcal{N} \rangle \equiv \sum_{\mathcal{N}=0}^{2^N} \mathcal{P}(\mathcal{N}) \mathcal{N} \geq \sum_{\mathcal{N}=1}^{2^N} \mathcal{P}(\mathcal{N}) = \mathbb{P}[\text{SAT}] \quad (3.3)$$

which means that the probability of having at least a solution is smaller than or equal to the average number of solutions. This bound for the probability that a formula is satisfiable is called *first moment inequality*.

Let us denote by $X \equiv \{x_i \mid i = 1, \dots, N\}$ a configuration of N boolean variables. In order to compute the average number of solutions of a random formula drawn from the uniform distribution, we introduce the indicator function $\varepsilon_l(X)$ which is equal to 1 if the configuration X verifies clause l and 0 otherwise. Then:

$$\langle \mathcal{N} \rangle = \left\langle \sum_X \prod_{l=1}^M \varepsilon_l(X) \right\rangle. \quad (3.4)$$

Since the clauses are extracted independently of one another, the average over the choices of the formula can be computed as an average over the choices of each clause appearing in it:

$$\langle \mathcal{N} \rangle = \sum_X \prod_{l=1}^M \langle \varepsilon_l(X) \rangle. \quad (3.5)$$

Moreover, the probability that *any* configuration X satisfies a uniformly drawn random clause is $1/2$, since for any choice of the indices appearing in the clause (and therefore, for fixed X , for any left hand side of the clause), the two choices TRUE and FALSE for the right hand side have equal probability. We obtain the very simple result:

$$\langle \mathcal{N} \rangle = 2^N \times 2^{-M} = 2^{N(1-\alpha)}. \quad (3.6)$$

and therefore from the first moment inequality:

$$\mathbb{P}[\text{SAT}] \leq \langle \mathcal{N} \rangle = 2^{N(1-\alpha)} \quad (3.7)$$

which goes to zero for $N \rightarrow \infty$ if $\alpha \geq 1$.

A lower bound for $\mathbb{P}[\text{SAT}]$ can be obtained from the *second moment inequality*, which is derived from the Cauchy-Schwarz inequality of the scalar product

$$\mathbf{u} \cdot \mathbf{v} \equiv \sum_{\mathcal{N}} \mathcal{P}(\mathcal{N}) u_{\mathcal{N}} v_{\mathcal{N}}, \quad (3.8)$$

which ensures that

$$(\mathbf{u} \cdot \mathbf{v})^2 \leq (\mathbf{u} \cdot \mathbf{u}) \times (\mathbf{v} \cdot \mathbf{v}) \quad (3.9)$$

for any vector \mathbf{u} and \mathbf{v} . In particular, by choosing $u_{\mathcal{N}} = \mathcal{N}$ for any \mathcal{N} and $v_{\mathcal{N}} = 1$ for $\mathcal{N} \geq 1$ and $v_0 = 0$ one obtains:

$$\langle \mathcal{N} \rangle^2 = \left(\sum_{\mathcal{N} \geq 1} \mathcal{P}(\mathcal{N}) \mathcal{N} \right)^2 \leq \left[\sum_{\mathcal{N} \geq 0} \mathcal{P}(\mathcal{N}) \mathcal{N}^2 \right] \times \left[\sum_{\mathcal{N} \geq 1} \mathcal{P}(\mathcal{N}) 1^2 \right] = \langle \mathcal{N}^2 \rangle \times \mathbb{P}[\text{SAT}]. \quad (3.10)$$

The crucial point is to compute

$$\langle \mathcal{N}^2 \rangle = \left\langle \left(\sum_X \prod_{l=1}^M \varepsilon_l(X) \right)^2 \right\rangle = \sum_{X,Y} \prod_{l=1}^M \langle \varepsilon_l(X) \varepsilon_l(Y) \rangle = \sum_{X,Y} \langle \varepsilon(X) \varepsilon(Y) \rangle^M \quad (3.11)$$

where again we made use of the independence of clauses in the extraction of a random formula to write the result in terms of $\langle \varepsilon(X) \varepsilon(Y) \rangle$ which is the probability that *both* X and Y satisfy a random clause. This quantity will obviously depend on how different X and Y are: if X satisfies the clause, Y will also satisfy it if and only if the number of variables appearing in the clause that are different in X and Y is even. When averaging over the choice of the clause, this will depend on the Hamming distance $d(X, Y)$ between X and Y ,

$$d(X, Y) \equiv \frac{1}{N} \sum_i \mathbb{I}(x_i \neq y_i). \quad (3.12)$$

For example, for $k = 3$ the probability that two configurations at distance d satisfy a random clause is

$$p_3(d) = \frac{1}{2} [(1-d)^3 + 3d^2(1-d)] + O(N^{-1}) \quad (3.13)$$

where the factor $1/2$ comes from the probability that X satisfies the clause to begin with; the term $(1-d)^3$ is the probability that the 3 variables appearing in the clause take the same value in X and Y ; the term $3d^2(1-d)$ is the probability that two variables are different and one is equal (among those appearing in the clause) in X and Y ; and finally, we are neglecting a term of order N^{-1} arising from the correlations in the choices of the variables appearing in a single clause (which must be different).

The general form will be

$$p_k(d) = \frac{1}{2} \sum_{l=0,2,\dots,k} \binom{k}{l} d^l (1-d)^{k-l} + O(N^{-1}) \quad (3.14)$$

in which only the *even* terms in the binomial expansion are taken. Notice that (contrary to the upper bound obtained from the first moment inequality), the lower bound derived from the second moment inequality will therefore depend on k .

Going back to (3.11) we obtain:

$$\langle \mathcal{N}^2 \rangle = \sum_{X,Y} p_k(d(X, Y))^M = \sum_{d=0, 1/N, 2/N, \dots} p_k(d)^M \mathcal{M}(d) \quad (3.15)$$

where $\mathcal{M}(d)$ is the number of pairs of configurations at distance d , i.e.

$$\mathcal{M}(d) = 2^N \binom{N}{Nd}, \quad (3.16)$$

so that for large N

$$\langle \mathcal{N}^2 \rangle = \sum_{d=0, 1/N, 2/N, \dots} \exp \{ N \log 2 [1 - (1-d) \log_2(1-d) - d \log_2 d + \alpha \log_2 p_k(d)] \} \quad (3.17)$$

which we evaluate with the Laplace method:

$$\langle \mathcal{N}^2 \rangle = 2^{N\gamma_k(\alpha, \bar{d})} \quad (3.18)$$

where $\gamma_k(\alpha, d)$ is the function multiplying $N \log 2$ in (3.17) and \bar{d} is the value of d that maximizes it in the interval $[0, 1]$. The result of the second moment calculation is:

$$\mathbb{P}[\text{SAT}] \geq \frac{\langle \mathcal{N} \rangle^2}{\langle \mathcal{N}^2 \rangle} = \frac{2^{2N(1-\alpha)}}{2^{N\gamma_k(\alpha, \bar{d})}} = 2^{N[2(1-\alpha) - \gamma_k(\alpha, \bar{d})]}. \quad (3.19)$$

For $k = 3$ one obtains that if $\alpha \leq \alpha_0(3) \simeq 0.889$ the function $\gamma_3(\alpha, d)$ has a global maximum in $\bar{d} = 1/2$ where $\gamma_3(\alpha, 1/2) = 2(1-\alpha) + o(1)$ (the asymptotics are for $N \rightarrow \infty$); if $\alpha > \alpha_0(3)$ a maximum located at $\bar{d} < 1/2$ becomes larger than the local one at $d = 1/2$ and $\gamma_3(\alpha, \bar{d}) > 2(1-\alpha) + o(1)$. Comparing with (3.19) one sees that, in the limit $N \rightarrow \infty$, $\mathbb{P}[\text{SAT}] > 0$ if $\alpha \leq \alpha_0(3)$.

The same analysis can be performed for larger values of k , leading to similar results. In fact, one can prove a stronger statement: not only $\mathbb{P}[\text{SAT}] > 0$ if $\alpha \leq \alpha_0(k)$, but the lower bound is equal to 1 in the thermodynamic limit, so that random k -XORSAT formulæ are SAT with probability 1 if $\alpha \leq \alpha_0(k)$.

The conclusion of the first and second moment calculations is that, if there is a sharp transition between the SAT and the UNSAT phases in k -XORSAT, it must occur for $\alpha = \alpha_s(k)$ such that

$$\alpha_0(k) \leq \alpha_s(k) \leq 1. \quad (3.20)$$

Since these bounds are not tight, one cannot conclude whether such a sharp transition exists on the basis of the first and second moment inequalities.

3.2.2 Leaf removal procedure

The leaf removal procedure allows to prove that a sharp transition between the SAT and UNSAT phases indeed exists, to compute the value of $\alpha_s(k)$ at which it occurs, and to characterize the geometry of the solutions [56, 57].

The main idea behind this powerful argument is the following: if the formula contains a variable x_1 which has a unique occurrence, the value of x_1 is constrained only by the clause in which it appears. Given the values of the other variables that appear in it, one is free to set the value of x_1 so as to satisfy the clause. This means that a clause which contains a single-occurrence variable does not constrain the values of the other variables that appear in it. One can then set it apart, and look for a solution of the reduced formula in which neither the single-occurrence variable nor the clause it belongs to are present. Moreover, when a clause is set apart, it is possible that some variable that appears in it becomes a single-occurrence variable (relative to the rest of the formula), so that the removal of single-occurrence variables (*leaves*) is an iterative procedure. In the following I shall give a quantitative description of this process.

Let us consider a random k -XORSAT formula with M clauses and N variables. It is easy to show that the distribution of the number of occurrences ℓ of the variables in the formula will be a poissonian with parameter αk :

$$\mathbb{P}[\ell] = e^{-\alpha k} \frac{(\alpha k)^\ell}{\ell!}. \quad (3.21)$$

A finite fraction $\alpha k e^{-\alpha k}$ of variables will therefore have a single occurrence. Let us assume that we proceed by removing them one at a time, in successive “steps”.

I shall denote by $n_\ell(T)$ the average number (divided by N) of variables that have ℓ occurrences after T steps. At that point, the total number of variables in the system is $N' = N - T$, and the

total number of clauses is $M' = M - T$, since at each step one variable and one clause are removed from the system. During a step, the number of occurrences of the other variables that appear in the removed clause will also be decreased by one. What is the probability that one of these other variables has ℓ occurrences? One might be tempted to say that it is just proportional to n_ℓ , since that is the probability that a variable has ℓ occurrences. However this is wrong, for the following reason. We can regard the formula as a table with M' rows and k columns, where the “slot” in row i and column j contains the index of the j^{th} variable in the i^{th} clause. A variable which has ℓ occurrences in the formula will appear in ℓ slots of the table. So the number of slots in the table that contain variables that have ℓ occurrences is $\ell \times N \times n_\ell$, and the probability that a randomly chosen slot contains a variable with ℓ occurrences is $\ell n_\ell / \sum_{\ell'} \ell' n_{\ell'}$. Since the number of variables in the removed clause is k , the average number of variables that appear in it (apart from the single-occurrence variable that we have chosen to eliminate) and that have ℓ occurrences is therefore $(k-1)\ell n_\ell / \sum_{\ell'} \ell' n_{\ell'}$.

We can use Wormald’s theorem, which I introduced in Chapter 2, to write a differential equation¹ for $n_\ell(t)$, where $t \equiv T/N$, in the limit $N \rightarrow \infty$:

$$\frac{dn_\ell}{dt} = (k-1) \frac{(\ell+1)n_{\ell+1}(t) - \ell n_\ell(t)}{k(\alpha-t)} \quad (\ell > 1) \quad (3.22)$$

where $k(\alpha-t) = \sum_{\ell} \ell n_\ell(t)$ is the total number of slots divided by N (remember that exactly k slots are removed at each step). The first term corresponds to the variables that have $\ell+1$ occurrences before the clause is removed, which afterwards have ℓ occurrences, while the second term corresponds to the variables that have ℓ occurrences before the clause is removed and which afterwards have $\ell-1$ occurrences. It is easy to check that this equation can be extended to $\ell=0$ and $\ell=1$ as follows:

$$\frac{dn_\ell}{dt} = (k-1) \frac{(\ell+1)n_{\ell+1}(t) - \ell n_\ell(t)}{k(\alpha-t)} + \delta_{\ell,0} - \delta_{\ell,1}. \quad (3.23)$$

The initial condition that must be imposed is (3.21)

$$n_\ell(0) = e^{-\alpha k} \frac{(\alpha k)^\ell}{\ell!}. \quad (3.24)$$

It is easy to see that, for $\ell \geq 2$, n_ℓ remains poissonian even for $t > 0$, with some time dependent parameter which is $\lambda(t)$. To prove it, one just needs to replace the ansatz

$$n_\ell(t) = e^{-\lambda(t)} \frac{\lambda(t)^\ell}{\ell!} \quad (3.25)$$

into (3.22) to obtain

$$\frac{dn_\ell}{dt} = -\frac{d\lambda}{dt} [n_\ell(t) - n_{\ell-1}(t)] = \frac{k-1}{k(\alpha-t)} \lambda(t) [n_\ell(t) - n_{\ell-1}(t)] \quad (3.26)$$

from which one obtains an equation for $\lambda(t)$ independent of ℓ :

$$\frac{d}{dt} \lambda(t) = -\frac{k-1}{k(\alpha-t)} \lambda(t). \quad (3.27)$$

Solving it with the initial condition $\lambda(0) = \alpha k$ gives

$$\lambda(t) = \alpha k \left(1 - \frac{t}{\alpha}\right)^{\frac{k-1}{k}}. \quad (3.28)$$

¹A detailed derivation is provided in Section 4.1 for a more general case.

However, $n_1(t)$ is *not* poissonian, because of the extra $\delta_{\ell,1}$ term in (3.23) compared to (3.22), and to compute it we use the following trick:

$$n_1(t) = \sum_{\ell=1}^{\infty} \ell n_{\ell}(t) - \sum_{\ell=2}^{\infty} \ell n_{\ell}(t) = k(\alpha - t) - \sum_{\ell=2}^{\infty} \ell e^{-\lambda(t)} \frac{\lambda(t)^{\ell}}{\ell!} = k(\alpha - t) - \left[\lambda(t) - \lambda(t)e^{-\lambda(t)} \right] \quad (3.29)$$

which can be conveniently expressed in terms of the parameter $b \equiv (1 - t/\alpha)^{1/k}$:

$$n_1(b) = \lambda(b) \left[b + e^{-\lambda(b)} - 1 \right] \quad (3.30)$$

with $\lambda(b) = \alpha k b^{k-1}$. The interval of variation of t is $[0, \alpha]$ (since after $\alpha N = M$ steps all the clauses are eliminated from the system), and correspondingly b varies between the initial value 1 and 0.

There are now two possibilities, depending on the value of α : either $n_1(b) > 0$ for all $b \in [0, 1]$, or for some value $b^* \in [0, 1]$ one has $n_1(b^*) = 0$. In the first case the algorithm stops when all the clauses have been removed from the system. In the second case, one is left with an irreducible sub-formula containing $N(\alpha - t^*) = N\alpha(b^*)^k$ clauses and $N \sum_{\ell=2}^{\infty} n_{\ell}(b^*) = N - N(1 - b^*)[1 + \alpha k(b^*)^{k-1}]$ variables. Note that the sub-formula is still uniformly random, conditioned on the distribution $n_{\ell}(b^*)$.

It is easy to check that the first case occurs for $\alpha < \alpha_c(k)$ where $\alpha_c(k)$ is a constant, while for $\alpha \geq \alpha_c(k)$ the value of n_1 vanishes for $b^* > 0$, which is the largest solution of (3.30). I shall denote $b_c(k)$ the value of b^* corresponding to $\alpha = \alpha_c(k)$. Numerical values of these constants (and their asymptotics for $k \rightarrow \infty$) are shown in Table 3.1, in the following paragraph.

Let us now turn to the implications of these results on the original formula. If the first case occurs (i.e. if $\alpha < \alpha_c(k)$), one can “invert” the procedure, and reinsert the clauses into the formula one at a time, in the reverse order with which they were removed. When the first clause is reinserted, one can choose freely the values of $k - 1$ variables, and set the last variable to the value which satisfies the clause. In general, when one reinserts a clause containing j “new” variables, the value of $j - 1$ of them is set arbitrarily, and the last one is set to the value which satisfies the clause. Notice that since each removed clause contained a variable which had a single occurrence at the time when it was removed, each reinserted clause will contain at least one new variable. One can then obtain a solution to the original formula in this manner.

What is the number of solutions that one obtains? Not counting the variable which has been selected for removal, the average number of single-occurrence (i.e. “new”) variables present in the clause removed at time t is $(k - 1)n_1(t)/[k(\alpha - t)]$. For each of them two values can be chosen. The number of solutions \mathcal{N} is therefore

$$\mathcal{N} \equiv 2^{Ns}, \quad s = \int_0^{t^*} \frac{(k - 1)n_1(t)}{k(\alpha - t)} dt + e^{-\alpha k} \quad (3.31)$$

where the last term comes from the variables which do not appear in the system. The integral is easily done recalling that for $\alpha < \alpha_c(k)$ one has $t^* = \alpha$ and substituting (3.28) and (3.29) to obtain $s = 1 - \alpha$, as expected from (3.6).

In the second case, for $\alpha_c(k) < \alpha$, the leaf removal procedure ends with a sub-formula with a clause to variables ratio α' given by

$$\alpha' = \frac{\alpha(b^*)^k}{1 - (1 - b^*)[1 + \alpha k(b^*)^{k-1}]}, \quad (3.32)$$

which is an increasing function of α . The original formula is SAT if and only if the sub-formula is also SAT, and we would like to know if this is the case, depending on the value of α' . As we have seen in

the calculation of the bound from the first and second moments (3.20), the upper bound $\alpha_s(k) \leq 1$ is independent on the distribution of random instances, while the lower bound $\alpha_0(k) \leq \alpha_s(k)$ depends on it. The computation of the lower bound must therefore be adapted to a distribution of instances which is uniform conditioned on the average numbers of occurrences $\{n_\ell(b^*)\}$ which is 0 for $\ell = 1$ and poissonian with parameter $\lambda(b^*)$ for $\ell \geq 2$. This is done in a detailed manner in [57]. The result is remarkable: in the absence of single-occurrence variables, the average number of solutions becomes a concentrated quantity and $\langle \mathcal{N}^2 \rangle = \langle \mathcal{N} \rangle^2$, so that the bounds from first and second moments inequalities become tight: $\alpha'_s(k) = 1$.

This proves that there is, indeed, a sharp transition between the SAT and the UNSAT phases, and the transition value of α is obtained from the condition

$$1 = \alpha' = \frac{\alpha(b^*)^k}{1 - (1 - b^*)[1 + \alpha k(b^*)^{k-1}]} \quad (3.33)$$

(notice that b^* is itself a function of α , determined by (3.30)).

The average number of solutions of the sub-formula will be

$$\mathcal{N}' = 2^{N'(1-\alpha')} = 2^N \{b^* - \alpha(b^*)^k + \alpha k[(b^*)^k - (b^*)^{k-1}]\}. \quad (3.34)$$

For each solution of the sub-formula, which I shall call “seed”, the number of solutions of the original formula that can be obtained is still given by (3.31), where now $t^* = \alpha[1 - (b^*)^k]$:

$$\mathcal{N}_1 = 2^N \{1 - b^* + \alpha k[(b^*)^{k-1} - (b^*)^k] - \alpha[1 - (b^*)^k]\} \quad (3.35)$$

where the subscript 1 is a reminder that this is for a fixed seed. Since for different seeds one necessarily obtains different solutions of the original formula, the total number of solutions is

$$\mathcal{N} = \mathcal{N}' \times \mathcal{N}_1 = 2^{N(1-\alpha)} \quad (3.36)$$

as expected.

It is possible to prove the following properties (or at least, to give some non-rigorous arguments supporting them, see [57, 58]):

1. The average distance d_0 between different solutions corresponding to the same seed is

$$d_0 = \frac{1 - b^*}{2} \quad (3.37)$$

2. The average distance d_1 between solutions corresponding to different seeds is

$$d_1 = \frac{1}{2} \quad (3.38)$$

3. The maximum distance between solutions corresponding to a same seed is smaller than the minimum distance between solutions corresponding to different seeds
4. For any two solutions X and X' corresponding to the same seed, there exists a sequence of solutions X_1, \dots, X_P such that $X_1 = X$, $X_P = X'$ and the (intensive) distance between X_i and X_{i+1} is of order $o(1)$ as $N \rightarrow \infty$.

k	$\alpha_c(k)$	$\alpha_s(k)$	$b_c(k)$
3	0.81846916	0.91793528	0.71533186
4	0.77227984	0.97677016	0.85100070
5	0.70178027	0.99243839	0.90335038
6	0.63708113	0.99737955	0.93007969
∞	$\log k/k$	$1 - e^{-k}$	$1 - 1/k \log k$

Table 3.1: Threshold values for the clustering and SAT/UNSAT transitions and backbone size b_c (at the clustering transition) for various values of k and (to the leading order) for $k \rightarrow \infty$.

3.2.3 Phase diagram of k -xorsat

Based on the previous analysis, the following phase diagram can be determined. Each statement is valid with probability 1 in the limit $N \rightarrow \infty$ for random k -XORSAT formulæ extracted from the uniform distribution and with $k \geq 3$.

The phase diagram of k -XORSAT consists of three phases, dependent on the ratio α of clauses per variable, separated by sharp transitions located at $\alpha_c(k)$ (for clustering) and $\alpha_s(k)$ (for SAT/UNSAT). Numerical values of the thresholds for finite k , and their asymptotics for $k \rightarrow \infty$ are shown in Table 3.1.

For $\alpha < \alpha_c(k)$ the formula is SAT and the solutions are homogeneously distributed in the space of configurations. Two random solutions are at an (intensive) distance $d = 1/2$, and they are connected by a sequence of solutions separated by a distance of order $o(1)$. The total number of solutions is given by (3.6),

$$\mathcal{N} = 2^{N(1-\alpha)}. \quad (3.39)$$

The value of the threshold $\alpha_c(k)$ is the smallest value of α such that the equation

$$b = 1 - e^{-\alpha k b^{k-1}} \quad (3.40)$$

has a solution with $b > 0$.

For $\alpha_c(k) < \alpha < \alpha_s(k)$, the formula is SAT and the solutions are clustered. Each cluster is identified by a particular solution of the sub-formula generated by the leaf-removal algorithm, called a seed. The solutions belonging to a same cluster are connected, the average distance between two of them is $(1 - b^*)/2$ and their number is given by (3.35):

$$\mathcal{N}_1 = 2^{N\{1-b^*+\alpha k[(b^*)^{k-1}-(b^*)^k]-\alpha[1-(b^*)^k]\}} \quad (3.41)$$

where b^* is the largest solution of (3.40), which represents the fraction of variables that take the same value in each solution of a given cluster and is called *back-bone* size. The solutions belonging to different clusters are well separated, the average distance between two of them is $1/2$ and the number of clusters is given by (3.34):

$$\mathcal{N}' = 2^{N'(1-\alpha')} = 2^{N\{b^*-\alpha(b^*)^k+\alpha k[(b^*)^k-(b^*)^{k-1}]\}}. \quad (3.42)$$

The threshold value $\alpha_s(k)$ is given by the condition (3.32):

$$\frac{\alpha(b^*)^k}{1 - (1 - b^*)[1 + \alpha k(b^*)^{k-1}]} = 1. \quad (3.43)$$

For $\alpha_s(k) < \alpha$ the formula is UNSAT. Note that as $\alpha \rightarrow \alpha_s(k)$ from below, the entropy of the number of cluster goes to 0, i.e. the number of clusters becomes sub-exponential in N , while the

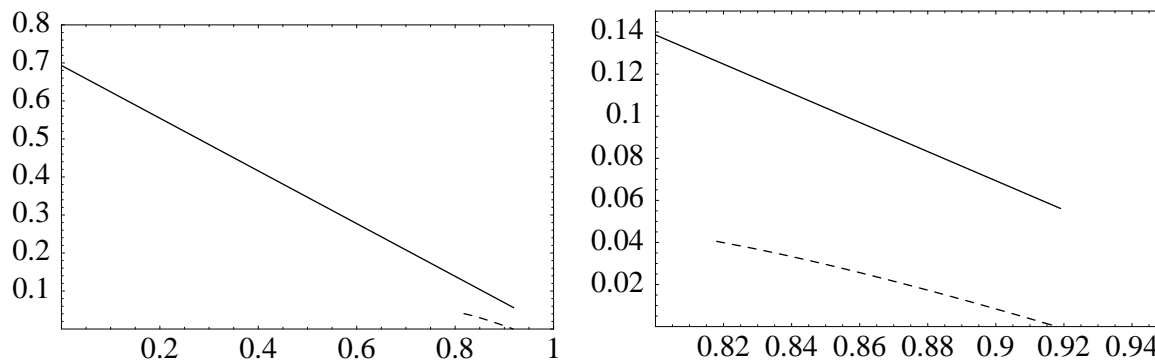


Figure 3.1: Total entropy $s(\alpha)$ (full line) and entropy of the number of clusters $\sigma(\alpha)$ (dashed line) as functions of α for 3-XORSAT. The curve for $\sigma(\alpha)$ starts at $\alpha = \alpha_c(3) \simeq 0.818$. The curve for $s(\alpha)$ ends at $\alpha = \alpha_s(3) \simeq 0.918$, where $\sigma(\alpha) = 0$. The right hand panel is an inset of the full figure, on the left.

number of solutions (inside each cluster) remains exponential in N , and discontinuously jumps to 0 as α crosses $\alpha_s(k)$.

The entropies (i.e. $\log \mathcal{N}/N$) of the number of clusters and of the (total) number of solutions are shown in Figure 3.1 for $k = 3$.

3.3 Heuristic results on the phase diagram of k -sat

The simple graph-theoretical arguments that allow the complete and rigorous characterization of the phase diagram of k -XORSAT do not apply in the case of k -SAT. Not only the methods required to derive it are more complicated (and not rigorous), but the phase diagram itself is more complicated.

Sat/Unsat transition

The existence of a SAT/UNSAT transition in k -SAT has been proved rigorously, but the proof of its sharpness remains an open problem. In fact, the following was proved by Friedgut [59]:

Theorem For each $k \geq 2$, there exists a sequence $\alpha_N(k)$ such that, for all $\epsilon > 0$,

$$\lim_{N \rightarrow \infty} \mathbb{P}[\text{Sat}|N, \alpha] = \begin{cases} 1 & \text{if } \alpha = (1 - \epsilon)\alpha_N(k), \\ 0 & \text{if } \alpha = (1 + \epsilon)\alpha_N(k) \end{cases} \quad (3.44)$$

where $\mathbb{P}[\text{Sat}|N, \alpha]$ is the probability that a uniformly random k -SAT formula with N variables and αN clauses is satisfiable.

Note that this theorem proves a non-uniform convergence: the threshold value is a function of N , which does not necessarily converge to a constant. This theorem doesn't imply that the SAT/UNSAT transition is sharp, but it proves that it exists. The sharpness of the transitions remains a conjecture.

Rigorous upper and lower bounds have been proved for the threshold $\alpha_s(k)$ for finite k and asymptotically as $k \rightarrow \infty$ (for a review and latest results, see [60]). Some values are listed in Table 3.2.

Finally, the best available estimates of the value of $\alpha_s(k)$ have been obtained with methods derived from statistical mechanics: the analysis of a message passing procedure called Survey Propagation (SP), which is based on the cavity method [61]. Some values obtained from the analysis of SP are reported in Table 3.2.

k	$\alpha_s^-(k)$	$\alpha_s^*(k)$	$\alpha_s^+(k)$
3	3.52	4.267	4.51
4	7.91	9.931	10.23
5	18.79	21.117	21.33
∞	$2^k \log 2 - k$	$2^k \log 2 - b_k$	$2^k \log 2$

Table 3.2: Threshold values for the SAT/UNSAT transition in k -SAT. $\alpha_s^-(k)$ is a rigorous lower bound, $\alpha_s^*(k)$ is the prediction from the cavity method, and $\alpha_s^+(k)$ is a rigorous upper bound. For $k \rightarrow \infty$ the rigorous bounds are exact, while in the result from the cavity computation, b_k is a positive function of k which converges to $(1 + \log 2)/2$ as $k \rightarrow \infty$. From [60, 61]

k	$\alpha_c(k)$	$\alpha_{\text{Cond}}(k)$	$\alpha_s(k)$
3	3.86	3.86	4.267
4	9.38	9.547	9.931
5	19.16	20.80	21.117

Table 3.3: Threshold values for the clustering (α_c) and condensation (α_{Cond}) transitions in k -SAT. The values of $\alpha_s(k)$ from Tab 3.2 are repeated for comparison. From [65].

Clustering transition

The satisfiable phase of k -SAT has a very rich structure, presenting *several* phase transitions that concern the geometry of the satisfying assignments. The first such transition is the clustering one.

The definition of the clustering phenomenon itself is much more complicated for k -SAT than for k -XORSAT. As we have seen, clustering in k -XORSAT has a geometrical origin: the set of variables of a formula can be decomposed in two: the backbone, made of variables that are determined by solving a sub-formula of the original problem; and the leaves, that are free to take any value in any solution. This structure naturally implies the clustering of solutions, and also two properties of the clusters: first, that all clusters contain the same number of solutions; second, that the variables that are frozen inside a cluster are the same for all clusters.

In k -SAT, these two properties do not hold. The fact that the variables that freeze in different clusters are not the same requires a definition of clusters independent on the backbone. This can be done by defining the clusters as a partition of the solutions such that:

1. The distance between any pair of solutions belonging to different clusters is larger than the distance between any pairs of solutions belonging to the same cluster;
2. For any pair of solutions (X, Y) belonging to the same cluster, a sequence of solutions $\{X_1, \dots, X_n\}$ can be made such that $X_1 = X$, $X_n = Y$ and the distance between X_i and X_{i+1} is of order $O(1)$ (as $N \rightarrow \infty$).

This approach is followed in [62, 63], where rigorous results are obtained for $k \geq 8$. Moreover, non-rigorous results based on the cavity method are available for any k [61], and are reported in Table 3.3.

Notice, however, that the clustering phenomenon was first suggested for k -SAT in [64], where a “variational” replica calculation was performed: based on physical intuition, a simple trial function with few free parameters was used as the functional order parameter for the free energy, as in (1.64),

and the values of the parameters were set by finding the extremum of the corresponding entropy. With this method, an approximation to the RSB solution which describes the clustered phase was found. This led to the calculation of approximate values of the clustering threshold $\alpha_c(k)$. In the same paper, the other difficulty arising in k -SAT, i.e. the fact that different clusters have different sizes, was pointed out.

It is a very important fact, as it gives rise to two more phase transitions.

Condensation and freezing transitions

Let us denote, as usual, the entropy of the number of clusters by Σ , the internal entropy of a cluster as s_i and the total entropy as s . Each of them is defined as the logarithm of the corresponding number of objects divided by N . When different clusters have different sizes, a convenient way of accounting for them is to write Σ as a function of s_i : $\Sigma(s_i)$ is the entropy of the number of clusters that have internal entropy s_i . The total entropy is then

$$s = \int [\Sigma(s_i) + s_i] ds_i. \quad (3.45)$$

The measure of the number of solutions will be dominated by the maximum of the integrand, i.e. by the value

$$s_i^* : \Sigma'(s_i^*) = -1. \quad (3.46)$$

At the clustering transition $\alpha_c(k)$, the complexity $\Sigma(s_i^*)$ becomes positive: the space of solutions splits into an exponential number of well separated clusters, each containing an exponential number of solutions. As α grows, the number of solutions decreases. More specifically, it is $\Sigma(s_i^*)$ which decreases, and for $\alpha = \alpha_{\text{Cond}}(k) < \alpha_s(k)$, it vanishes. When this happens, both the number of solutions and the number of cluster are still exponential; however, the measure of the number of solutions is dominated by a sub-exponential number of clusters, corresponding to the largest s_i . As α increases further, the value of the *maximum* of $\Sigma(s_i)$ decreases, until it vanishes at $\alpha = \alpha_s(k)$, the SAT/UNSAT transition. When this happens, the number of solutions vanishes abruptly, with a discontinuity in s_i .

Still another phase transition occurs for intermediate values of α , corresponding to the freezing of variables within a cluster. For $\alpha_c(k) < \alpha < \alpha_f(k)$, there are no frozen variables (even within a cluster), while for $\alpha_f(k) < \alpha$ frozen variables are present [66].

Part II

Some properties of random k -SAT and random k -XORSAT

Chapter 4

Study of poissonian heuristics for DPLL in k -XORSAT

In this chapter I shall present some new results on the relationship between the clustering transition of k -XORSAT and the performance of DPLL algorithms, obtained with Rémi Monasson and Francesco Zamponi and published in [67].

It is generally believed that *local* algorithms cannot succeed (in finding solutions) in the clustered phase of random CSP. In this context “local” means that the algorithm decides assignments based on local information, such as the values of variables within a finite subset of clauses. Local algorithms therefore include, for example, search algorithms such as Metropolis or WalkSAT, and also the DPLL procedure. The basic argument supporting this belief is that in the clustered phase an extensive back-bone of frozen variables exists, which requires an extensive number of variables to take values that are strongly correlated. An optimization procedure which only takes into account a finite portion of the problem will not be able to find a correct assignment for the back-bone, and therefore for the problem.

An alternative argument is directly derived from spin glass theory: the free energy landscape of random CSP in the clustered phase is characterized by a large number of states, most of which have positive energy, separated by extensive barriers. In order to go from a random configuration to a ground state the system must cross these barriers, which a local optimization procedure cannot do. If this argument is plausible for search procedures, which perform a random walk in the space of configurations while trying to minimize some cost function, and which therefore can indeed remain trapped in local minima of the free energy, it is not at all clear why it should apply to the DPLL procedure. Indeed, the only evidence supporting this claim for DPLL is that no heuristic is known to succeed in the clustered phase.

The main result that I shall present in this chapter is that no DPLL heuristic which preserves the poissonian distribution of occurrences in the sub-formulæ it generates can find solutions in the clustered phase. The essence of the argument, as we shall see, is related to the geometrical properties of the graph underlying the formula (which allow the use of the leaf-removal procedure to characterize the phases), and to the very basic fact that a Unit Propagations cannot remove more than one clause for each variable that they assign.

This result is valid for random k -XORSAT formulæ extracted from the uniform distribution (with probability 1 as $N \rightarrow \infty$, as usual). It is worth noting that it can be extended to a generalization of k -XORSAT which goes under the name of Uniquely Extensible Constraint Satisfaction Problems,

or UE-CSP. In these problems, variables can take values in a set of cardinality d , and the form of the constraints is such that k variables appear in them, and that if any $k - 1$ variables appearing in a constraint are assigned, then the value of the k^{th} variable is determined. It is very interesting that (d, k) -UE-CSP is NP-Complete for $\{d \geq 4, k \geq 3\}$. k -XORSAT is a special case of (d, k) -UE-CSP with $d = 2$. However, as far as the DPLL procedure is concerned, the class of (d, k) -UE-CSP is equivalent to k -XORSAT for *any* k and d , since the only relevant feature for the sake of DPLL is that Unit Propagations be possible, and the characteristic property of UE-CSP's ensures that it is. As a consequence that there will be a sharp transition between a phase with a back-bone and a phase without it, which will occur for some $\alpha_c(d, k)$, and that all the results that we shall derive concerning the performance of DPLL will be valid for (d, k) -UE-CSP as well.

The structure of this chapter is the following: in Section 4.1 I shall introduce a generalization of the leaf-removal procedure to mixed formulæ; this will allow me to introduce a *potential* function that characterizes the phases of mixed formulæ, in Section 4.2; in Section 4.3 I shall characterize the trajectories that poissonian heuristics generate in the space of the density of clauses $\{c_j\}$; then I shall derive an upper bound for the values of α for which poissonian heuristics for DPLL can find solutions, in Section 4.4; in Section 4.5 I shall present an argument supporting that GUC saturates the previous bound in the limit $k \rightarrow \infty$. ; finally, in Section 4.6 I shall discuss the results obtained and indicate some possible directions of further investigation.

4.1 Leaf-removal for mixed formulæ

In paragraph 3.2.2 I described the leaf-removal procedure applied to a *pure* k -XORSAT formula, that is to say a formula in which all the clauses involve exactly k variables, as was introduced in [56, 57]. The leaf-removal procedure is extremely powerful, as it provides a full characterization of the phase diagram of k -XORSAT. In this Section I shall generalize the analysis of the leaf-removal procedure to the case of *mixed* formulæ, containing clauses of *different lengths* (where length stands for the number of variables in the clause), in order to allow the characterization of the sub-formulæ generated by DPLL heuristics.

4.1.1 Leaf-removal differential equations

Let us consider a random XORSAT formula with N variables and a total of M clauses of different lengths $j = 2, 3, \dots, k$. We don't consider clauses of length 1 since they are trivial, and we denote by k the maximum clause length. The number of clauses of length j will be denoted by $C_j(0)$, where the 0 indicates that this is the initial formula (relative to the action of the leaf-removal), and we will have $M = \sum_{j=2}^k C_j(0) = \alpha N$ for some finite α . We shall also denote by $N_\ell(0)$ the number of variables with ℓ occurrences, and therefore $\sum_{\ell=0}^{\infty} N_\ell(0) = N$. We assume that the formula is formed by selecting uniformly at random the index of the variable appearing in each "slot" of each clause (with no repetitions within a clause). The distribution of the number of occurrences of the variables in the formula is then a poissonian with parameter $\lambda(0)$. Notice that the distribution of occurrences is *independent* on the clause lengths (i.e. the distribution of occurrences in clauses of length j is the same for all j).

The leaf-removal proceeds in steps. Let us denote by T the number of steps that have been performed. At each step, a single-occurrence variable is selected, and the clause in which it appears is removed. What is the probability $p(j)$ that a single occurrence variable appears in a clause of length j ? By definition, a single-occurrence variable occupies a unique slot in the formula. Since each slot

can contain any variable with uniform probability, $p(j)$ will be proportional to the fraction of slots that belong to clauses of length j :

$$p(j) = \frac{jC_j}{\sum_j jC_j}. \quad (4.1)$$

If we denote by $C_j(T)$ the number of clauses of length j after T steps of leaf-removal, we shall have

$$\mathbb{E}[C_j(T+1) - C_j(T)] = -p(j) = -\frac{jC_j(T)}{\sum_j jC_j(T)}. \quad (4.2)$$

Moreover, if the removed clause has length j , the average number of variables (excluding the one to be eliminated) with ℓ occurrences that appear in it will be $(j-1)\ell N_\ell(T) / \sum_{\ell'} \ell' N_{\ell'}$, and therefore

$$\mathbb{E}[N_\ell(T+1) - N_\ell(T)|j] = (j-1) \frac{(\ell+1)N_{\ell+1}(T) - \ell N_\ell(T)}{\sum_{\ell'} \ell' N_{\ell'}(T)} + \delta_{\ell,0} - \delta_{\ell,1} \quad (4.3)$$

where the Kronecker deltas come from the single occurrence variable being eliminated. Multiplying by $p(j)$ and summing over j ,

$$\mathbb{E}[N_\ell(T+1) - N_\ell(T)] = \sum_{j=2}^k p(j) \mathbb{E}[N_\ell(T+1) - N_\ell(T)|j] \quad (4.4)$$

$$= \sum_{j=2}^k \frac{jC_j(T)}{\sum_j jC_j(T)} (j-1) \frac{(\ell+1)N_{\ell+1}(T) - \ell N_\ell(T)}{\sum_{\ell'} \ell' N_{\ell'}(T)} + \delta_{\ell,0} - \delta_{\ell,1}. \quad (4.5)$$

In the limit $N \rightarrow \infty$ the variations in (4.2) and (4.5) are of $O(1)$, and we can apply Wormald's theorem to obtain the following differential equations for $n_\ell(t) \equiv \mathbb{E}[N_\ell(Nt)/N]$ and $c_j(t) \equiv \mathbb{E}[C_j(Nt)/N]$:

$$\begin{cases} \frac{dc_j(t)}{dt} = -\frac{j c_j(t)}{\sum_{j'=2}^k j' c_{j'}(t)}, \\ \frac{dn_\ell(t)}{dt} = \sum_{j=2}^k \frac{j(j-1)c_j(t)}{\sum_{j'=2}^k j' c_{j'}(t)} \frac{(\ell+1)n_{\ell+1}(t) - \ell n_\ell(t)}{\sum_{\ell'=0}^\infty \ell' n_{\ell'}(t)} + \delta_{\ell,0} - \delta_{\ell,1}. \end{cases} \quad (4.6)$$

The initial conditions for $c_j(t)$ are trivial (i.e. $c_j(0) = C_j(0)/N$), while those for n_ℓ are:

$$n_\ell(0) = e^{-\lambda(0)} \frac{\lambda(0)^\ell}{\ell!}. \quad (4.7)$$

Since the parameter of the poissonian coincides with its average, we shall have

$$\lambda(0) = \sum_{\ell=0}^\infty \ell n_\ell(0) = \sum_{j=2}^k j c_j(0) \quad (4.8)$$

where the last equality comes from the fact that the two sums in (4.8) give the number of slots in the formula, and therefore are equal.

4.1.2 Solution for $c_j(t)$

In order to solve (4.6), we observe two things: first, that the equations for $\{c_j(t)\}$ are independent on $\{n_\ell(t)\}$; second, that the equation for $c_k(t)$ implies that, as long as $c_k(t) > 0$, it is a strictly decreasing function of t , and therefore c_k can be used as an independent variable instead of t . We then divide the equations for c_j with $j = 2, 3, \dots, k-1$ by the equation for c_k , obtaining

$$\frac{dc_j}{dc_k} = \frac{j}{k} \frac{c_j}{c_k} \quad (4.9)$$

which admits the solution

$$c_j = c_j^0 \left(\frac{c_k}{c_k^0} \right)^{j/k} \quad (4.10)$$

where c_j^0 is the value of c_j at $t = 0$. It is convenient to introduce

$$b(t) \equiv \left(\frac{c_k(t)}{c_k(0)} \right)^{1/k} \quad (4.11)$$

so that (4.10) becomes

$$c_j(t) = c_j(0)b(t)^j. \quad (4.12)$$

Notice that b is an invertible function of c_k , and therefore of t .

Let us also introduce the generating function $\gamma(b)$ of the $c_j(0)$, which will play a very important role in the following:

$$\gamma(b) \equiv \sum_{j=2}^k c_j(0)b^j \quad (4.13)$$

so that

$$\gamma(b(t)) = \sum_{j=2}^k c_j(0)b(t)^j = \sum_{j=2}^k c_j(t) \quad (4.14)$$

is the total number of clauses at time t and

$$b(t)\gamma'(b(t)) \equiv b(t) \left. \frac{d\gamma}{db} \right|_{b=b(t)} = \sum_{j=2}^k j c_j(t) = \sum_{\ell=1}^{\infty} \ell n_{\ell}(t) \quad (4.15)$$

is the number of slots in the formula at time t . Since exactly one equation is removed at each step, one must have

$$\gamma(b(t)) = \alpha - t \quad (4.16)$$

which implicitly defines $b(t)$ through (4.13):

$$t(b) = \alpha - \sum_{j=2}^k c_j(0)b^j. \quad (4.17)$$

4.1.3 Solution for $n_{\ell}(t)$

We can now write the equations for $\{n_{\ell}(t) \mid \ell \geq 2\}$ in (4.6) as

$$\frac{dn_{\ell}}{dt} = \frac{\gamma''(b)b^2}{[\gamma'(b)b]^2} \Big|_{b=b(t)} [(\ell+1)n_{\ell+1} - \ell n_{\ell}] = \frac{\gamma''(b)}{\gamma'(b)^2} \Big|_{b=b(t)} [(\ell+1)n_{\ell+1} - \ell n_{\ell}]. \quad (4.18)$$

As in the case of pure k -XORSAT formulæ the distribution of occurrences (for $\ell \geq 2$) remains poissonian at all times:

$$n_{\ell}(t) = e^{-\lambda(t)} \frac{\lambda(t)^{\ell}}{\ell!} \quad (\ell \geq 2). \quad (4.19)$$

This is easily seen by substituting this expression in (4.18), which gives an equation for λ which is independent on ℓ :

$$\frac{d\lambda}{dt} = -\frac{\gamma''(b)}{\gamma'(b)^2} \lambda \quad (4.20)$$

where $b = b(t)$. This is solved by noticing from (4.17) that

$$\frac{dt}{db} = -\gamma'(b) \quad (4.21)$$

so that

$$\frac{d\lambda}{db} = \frac{\gamma''(b)}{\gamma'(b)} \lambda \quad (4.22)$$

with the initial condition that for $t = 0$, which corresponds to $b = 1$, λ must be equal to $\lambda(0) = \sum_j j c_j(0)$. The solution is then:

$$\lambda(b) = \gamma'(b) \quad (4.23)$$

and we obtain

$$n_\ell(b) = e^{-\gamma'(b)} \frac{\gamma'(b)^\ell}{\ell!} \quad (\ell \geq 2) \quad (4.24)$$

with $b = b(t)$ obtained by inverting $t = \alpha - \gamma(b)$.

For $\ell = 1$ we write

$$n_1(b) = \sum_{\ell=1}^{\infty} \ell n_\ell(b) - \sum_{\ell=2}^{\infty} \ell n_\ell(b) = b\gamma'(b) - e^{-\gamma'(b)} \gamma'(b) \left[e^{\gamma'(b)} - 1 \right]. \quad (4.25)$$

The leaf-removal will end when $n_1(b) = 0$ for some $b \in [0, 1]$, which gives:

$$b = 1 - e^{-\gamma'(b)}. \quad (4.26)$$

Let us denote by b^* the largest solution of this equation. If $b^* = 0$ the leaf-removal removes all the clauses from the formula, which is SAT (with probability 1), and the solutions are unclustered. If $b^* > 0$ the leaf-removal ends with an irreducible sub-formula. The number of clauses in the sub-formula is $N \sum_{j=2}^k c_j(t^*) = N\gamma(b^*)$ and the number of variables is $N \sum_{\ell=2}^{\infty} n_\ell(b^*) = N e^{-\gamma'(b^*)} \left[e^{\gamma'(b^*)} - 1 - \gamma'(b^*) \right]$. The sub-formula is SAT if and only if the number of variables is smaller than or equal to the number of clauses:

$$\gamma(b^*) \leq b^* + (1 - b^*) \log(1 - b^*) \quad (4.27)$$

where we have used (4.26). The SAT/UNSAT transition occurs when this bound is saturated.

4.2 Characterization of the phases in terms of a potential

4.2.1 Definition and properties of the potential $V(b)$

Let us define the following *potential*, which is a function of b :

$$V(b) = -\gamma(b) + b + (1 - b) \log(1 - b). \quad (4.28)$$

The derivative of $V(b)$ is

$$V'(b) = -\gamma'(b) - \log(1 - b), \quad (4.29)$$

and we see that for $b = b^*$, which verifies (4.26) $\Leftrightarrow \gamma'(b^*) = -\log(1 - b^*)$, we have

$$V'(b^*) = 0. \quad (4.30)$$

The value of b^* can therefore be obtained from $V(b)$, looking for the largest value in $[0, 1]$ where the derivative of V vanishes.

In the unclustered phase, $V(b)$ has a unique minimum at $b^* = 0$. As α grows, a secondary minimum develops for $b^* > 0$. The clustering transition occurs when this secondary minimum forms, and when this happens one must have $V''(b^*) = 0$. On the other hand, from (4.27) and (4.28) one sees that the

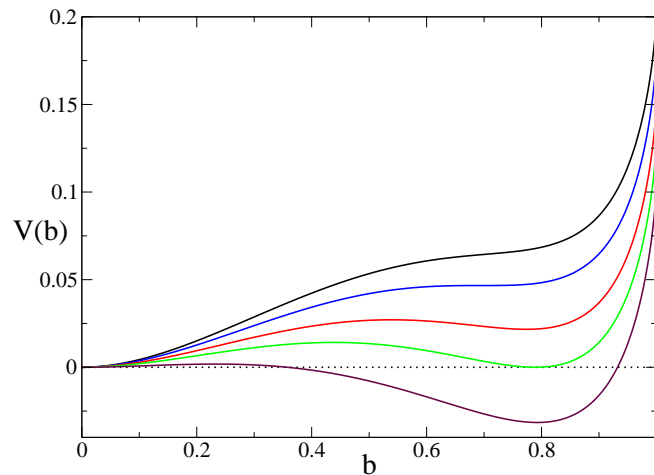


Figure 4.1: Potential $V(b)$ for different formulæ. Each was obtained by applying the UC heuristic to a 3-XORSAT formula with $\alpha = 0.8$ for different times: from top to bottom $t = \{0, t_c = 0.02957, 0.07327, t_s = 0.11697, 0.20642\}$. The first curve shows that the formula is in the unclustered phase; the second curve corresponds to the clustering transition; the third to a clustered formula; the fourth to the SAT/UNSAT transition; finally, the formula is UNSAT.

SAT/UNSAT transition occurs when $V(b^*) = 0$. As in the pure case of paragraph 3.2.2, b^* is the size of the back-bone, i.e. the fraction of variables that take the same value in each solution of a given cluster.

It is therefore possible to characterize the phase to which the formula belongs in terms of $V(b)$:

$$\text{Back-bone size:} \quad b^* = \max_{b \in [0,1]} \{b : V'(b) = 0\} \quad (4.31)$$

$$\text{Clustering transition:} \quad V''(b^*) = 0 \quad (4.32)$$

$$\text{SAT/UNSAT transition:} \quad V(b^*) = 0 \quad (4.33)$$

An example of potential is provided in Figure 4.1. The formulæ considered for each curve are generated by the UC heuristic applied to a 3-XORSAT formula with $\alpha = 0.8$. Each curve corresponds to a different time during the evolution under UC (more detailed explanations are given in Section 4.3).

Notice that, given an arbitrary set of clause densities $\{c_2, \dots, c_k\}$, it is not *a priori* a trivial task to determine whether random formulæ conditioned by $\{c_j\}$ are SAT or not, and if they are SAT, whether their solutions are clustered or not. However, it suffices to compute $V(b)$ for the given set of c_j 's and, from its “shape”, the answers to the previous questions become immediately clear. This is what makes the potential $V(b)$ such a powerful tool in the study of the phase transitions of k -XORSAT (and of (d, k) -UE-CSP).

Interestingly, a “potential method” was already well known in mean field theory of spin glasses. It was originally introduced by Parisi [68] and developed by him and Franz [69, 70, 71, 72] and by Monasson [73]. “Their” potential is derived in a completely different way: it is defined by considering two *real* replicas of the system (i.e. two identical samples), with an interaction term that depends on the overlap q between their configurations. The first replica is allowed to equilibrate at temperature T , without “feeling” the effect of the coupling, while the second replica equilibrates at the same temperature but is subject to the interaction. The effect of the interaction is to constrain the configurations

of the second replica to those that have a fixed overlap with the equilibrium configurations of the first one. The potential $V(q)$ is then defined as the free energy of the second replica as a function of q , in the limit in which the interaction strength vanishes.

Even though the potential $V(q)$ is defined in a completely different way from the potential $V(b)$ defined in this Section, the two share many common features. First, both are functions of the overlap q , or equivalently of the fraction of frozen spins b ; second, the properties of their minima determine the phase transitions of the system (in this regard, the definition of $V(q)$ as a free energy is much more transparent); third, the value of the potential corresponding to the secondary minimum (when it is present) is equal to the complexity. In fact, it should be possible to prove that the two potentials are actually identical by computing the full expression of the 1RSB free energy of k -XORSAT in the case of a mixed system, and deriving the explicit expression of $V(q)$ in the most general case. The fact that the same potential can be obtained following two approaches that are so different is a very interesting fact in itself.

4.2.2 Phase diagram for mixed k -xorsat formulæ

For pure formulæ the phase diagram depends on a single parameter, α . For mixed formulæ, the phase diagram is more complicated, as the space of parameters is $\mathcal{C} = \{c_2, \dots, c_k\}$ which has dimension $k - 1$. Each one of the c_j 's varies in $[0, 1]$, because if some $c_{j'} > 1$ then the sub-formula containing only the clauses of length j' is UNSAT (and therefore so is the complete formula). For any point $\mathbf{c} \in \mathcal{C}$ we can compute the potential $V(b)$, which depends on \mathbf{c} through $\gamma(b) = \sum_{j=2}^k c_j b^j$, and we can define b^* as the largest solution of $b = 1 - e^{-\gamma'(b)}$ in $[0, 1]$.

The phase transitions are characterized by the conditions (4.32) and (4.33). The boundary between the unclustered and the clustered phase will be the $(k - 2)$ -dimensional surface Σ_c defined by:

$$\Sigma_c = \{\mathbf{c} \in \mathcal{C} : (b^* > 0) \wedge (V''(b^*) = 0)\} . \quad (4.34)$$

The boundary between the SAT and the UNSAT regions in \mathcal{C} will be the $(k - 2)$ -dimensional surface Σ_s defined by:

$$\Sigma_s = \{\mathbf{c} \in \mathcal{C} : (b^* > 0) \wedge (V(b^*) = 0)\} . \quad (4.35)$$

Notice that in $b = 0$ one always has $V(0) = V'(0) = 0$, because the first term in $\gamma(b)$ is $c_2 b^2$ and $b + (1 - b) \log(1 - b) = b^2/2 + O(b^3)$ for small b . Also, for $c_2 = 1/2$ one has $V''(0) = 0$ (irrespectively of the values of c_j for $j > 2$). Therefore, for $c_2 = 1/2$, $b = 0$ is formally a solution of $V(b) = 0$ and of $V''(b) = 0$. Even though the surfaces Σ_c and Σ_s are defined with $b^* > 0$, it is possible that $b^* \rightarrow 0$ if the local minimum at $b^* > 0$ merges into the global minimum of V in $b = 0$. This can happen if and only if $V'''(0) = 0$ (so that $b = 0$ becomes a “flat” saddle of V), which is obtained for $c_3 = 1/6$ (as seen by taking the term in b^3 in the above expansions). This implies that the two surfaces Σ_c and Σ_s intersect on the $(k - 3)$ -dimensional surface Σ_k of equation:

$$\Sigma_k = \left\{ \mathbf{c} \in \mathcal{C} : \left(c_2 = \frac{1}{2} \right) \wedge \left(c_3 = \frac{1}{6} \right) \right\} \quad (4.36)$$

The suffix ‘k’ stands for *critical* (the ‘c’ being used for *clustering*), because Σ_k is the surface where the discontinuous phase transitions (both the clustering and the SAT/UNSAT ones) become continuous, which is traditionally called *critical point* in statistical mechanics.

The surfaces Σ_c , Σ_s and Σ_k are tangent to each other. This can be seen by verifying that $c_2 = 1/2 - \epsilon^2$ and $c_3 = 1/6 + \epsilon$ verify (4.34), (4.35) and (4.36) with $b^* = \epsilon$ (to the leading order in $\epsilon \rightarrow 0$).

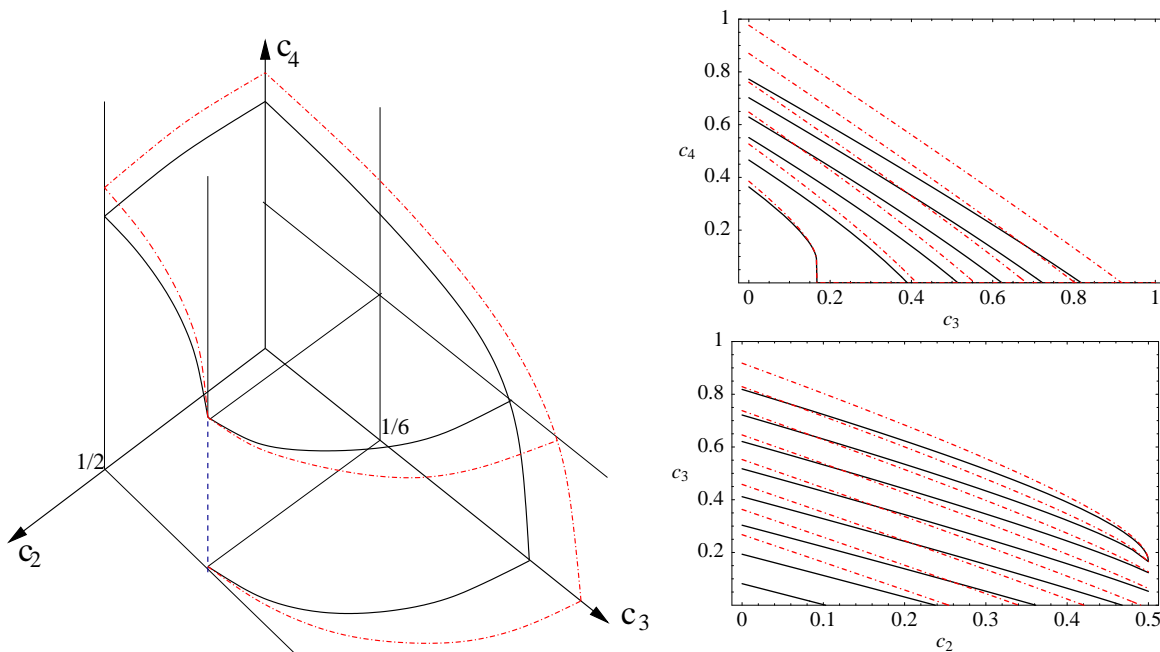


Figure 4.2: Phase diagram of mixed 4-XORSAT. *Left* A pictorial view of the surfaces Σ_c (full black) and Σ_s (dot-dashed red), intersecting on the segment Σ_k (dashed blue), where they are tangent to each other. Going from the origin out, the formulæ are first unclustered, then clustered (after Σ_c is crossed) and finally UNSAT (after Σ_s is crossed). *Right* The sections of Σ_c (full black) and Σ_s (dot-dashed red) at constant $c_2 = \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ from top to bottom (top panel) and at constant $c_4 = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ from top to bottom (bottom panel). The phase diagram for pure formulæ with $k = 3$ is formed by the c_3 axis of the bottom panel, which the two curves corresponding to $c_4 = 0$ intersect at $c_3 = \alpha = 0.818$ and $c_3 = \alpha = 0.918$ respectively.

The fact that Σ_c and Σ_s have an intersection where they are tangent to each other is not at all clear *a priori* (as we have seen, it depends on the specific form of $V(b)$), but it will turn out to be extremely important in the following.

As an illustration, the phase diagram for $k = 4$ is shown in Figure 4.2.

4.3 Trajectories generated by poissonian heuristics

In Section 2.4 I introduced the DPLL procedure and discussed some properties of two specific heuristics, called respectively Unit Clause (UC) and Generalized Unit Clause (GUC), for the problem of $(2 + p)$ -XORSAT. In this section I shall extend the same kind of analysis to more general heuristics and to mixed formulæ of any maximum length k .

I shall first define the class of heuristics considered, then derive some general properties of poissonian heuristics that will be useful in Section 4.4, and finally analyze the special cases of UC and GUC to illustrate them.

4.3.1 Poissonian heuristics for DPLL

Let us consider a DPLL procedure without back-tracking acting on some pure k -XORSAT formula. For the class of heuristics I want to introduce, it is convenient to modify the description of the procedure I gave in Section 4.4 in such a way that unit propagations are performed by the heuristic. The modified procedure is described by the following pseudocode:

```

procedure MODIFIED DPLL( $\{C_2(0), \dots, C_k(0)\}$ )
  repeat
    Select and assign a variable  $x$  according to HEURISTIC
    Simplify the formula
  until A contradiction is generated or All the variables are assigned
end procedure

```

with the heuristic:

```

procedure POISSONIAN HEURISTIC( $\{p_j(C_1, \dots, C_k) \mid j = 0, \dots, k\}$ )
  switch With probability  $p_0(C_1, \dots, C_k)$ :
    Select uniformly at random a variable  $x$ 
    Assign  $x$  to TRUE or FALSE uniformly at random
  otherwise With probability  $1 - p_0(C_1, \dots, C_k)$ :
    Select at random a clause length  $j \in \{1, \dots, k\}$  with probability  $p_j(C_1, \dots, C_k)$ 
    Select uniformly at random a clause  $\mathcal{C}$  of length  $j$ 
    Select uniformly at random a variable  $x$  appearing in  $\mathcal{C}$ 
    Assign  $x$  to TRUE or FALSE uniformly at random
  end switch
end procedure

```

where $p_j(C_1, \dots, C_k)$ with $j = 0, \dots, k$ are functions that characterize the heuristic. The Unit Propagation rule then simply requires that $p_j(\{C_j\}) = \delta_{j,0}$ if $C_1 > 0$. Notice that $\{C_j\}$ are the extensive numbers of clauses of length j in the specific formula we are considering (they are not averaged over the distribution of formulæ). Moreover, since the alternatives corresponding to different values of $j = 0, \dots, k$ are independent, it is possible to normalize the probabilities so that

$$\sum_{j=0}^k p_j(C_1, \dots, C_k) = 1. \quad (4.37)$$

It is easy to see that UC and GUC are special cases of this class of heuristics:

$$p_j^{\text{UC}}(\{C_j\}) = \begin{cases} \delta_{j,1} & \text{if } C_1 > 0; \\ \delta_{j,0} & \text{otherwise.} \end{cases} \quad (4.38)$$

$$p_j^{\text{GUC}}(\{C_j\}) = \begin{cases} \delta_{j,1} & \text{if } C_1 > 0; \\ \mathbb{I}[j \text{ is the length of the shortest clause in the formula}] & \text{otherwise.} \end{cases} \quad (4.39)$$

A very important property of this class of heuristics is that the sub-formulæ that it generates are uniformly distributed, conditioned on the numbers $\{C_j\}$ of clauses of length j . As a consequence, the distribution of the number of occurrences of variables will remain poissonian under the action of these heuristics, even though the parameter of the poissonian may vary. This is the reason why I call this class of heuristics *poissonian*. In fact, I believe this to be the most general class of heuristics which preserve the uniform distribution of the sub-formulæ it generates (even though I am unable to support this claim).

Because of this property of the heuristics it is possible to analyze them in terms of differential equations, as we did for UC and GUC in Section 2.4. We define the *time* $t = T/N$ where T is the number of variables that have been assigned, and the average clause densities $c_j(t) = \mathbb{E}[C_j(Nt)/N]$. The initial condition for the equations will be $c_j(0) = \alpha\delta_{j,k}$. Under the action of the heuristic, the formula will trace a trajectory in the space $\{c_j\} \subset [0, 1]^{k-1}$. The dimension of the space is $k - 1$ instead of k because if at any time $c_1(t) > 0$ the procedure generates a contradiction with probability 1 and it fails.

For notational convenience, I shall introduce $c_{k+1}(t) \equiv 0$ and $p_{k+1}(\{C_j\}) \equiv 0$. An analysis similar to that carried out in Section 2.4 for GUC then shows that the differential equations that determine $\{c_j\}$ are the following:

$$\frac{dc_j}{dt} = \frac{(j+1)c_{j+1}(t) - jc_j(t)}{1-t} - \rho_j(t) \quad (j = 1, \dots, k) \quad (4.40)$$

where

$$\rho_j(t) \equiv \lim_{\Delta T \rightarrow \infty} \frac{1}{\Delta T} \lim_{N \rightarrow \infty} \sum_{T=tN}^{tN+\Delta T-1} [p_j(\{C_{j'}(T)\}) - p_{j+1}(\{C_{j'}(T)\})] \quad (j = 1, \dots, k) \quad (4.41)$$

is (minus) the average variation of c_j due to the the algorithm selecting $j+1$ or j as the length for the clause from which to pick the variable to be assigned. In this equation ΔT is a number of steps of order $o(N)$, so that $c_j(t)$ can be considered constant over ΔT , and which is a generalization of the “round” I introduced in the analysis of GUC. Notice that $\rho_j(t)$ depends on t only through $\{c_j(t)\}$.

The first term in (4.40) is due to the other clauses of the formula in which the selected variable appears: on average, there will be $(j+1)c_{j+1}/(1-t)$ of them of length $j+1$ (which will become of length j) and $jc_j(t)/(1-t)$ of length j (which will become of length $j-1$).

Since the density of unit clauses in the formula is always 0, for $j = 1$ (4.40) reduces to

$$\frac{dc_1}{dt} = \frac{2c_2(t)}{1-t} - \rho_1(t) = 0 \quad (4.42)$$

which gives the explicit expression of $\rho_1(t)$ required to ensure Unit Propagation. The condition that signals the appearance of contradictions with probability 1 is

$$\rho_1(t) = \frac{2c_2(t)}{1-t} = 1. \quad (4.43)$$

I shall define one more $(k-2)$ -dimensional surface in the phase diagram:

$$\Sigma_q = \left\{ \tilde{\mathbf{c}} \in [0, 1]^{k-1} : \tilde{c}_2 = \frac{1}{2} \right\} \quad (4.44)$$

where the ‘q’ stands for *contradiction* (the ‘c’ being very much in demand...) and where the tilde reminds us that these clause densities are normalized to the number of variables in the sub-formula, i.e. $\tilde{c}_j = c_j/(1-t)$.

A final remark to conclude this paragraph: since the distribution of occurrences remains poissonian at all times, the results of the previous section allow to characterize the phase to which the sub-formulae generated by the heuristics belong. The only difference is that the clause densities $c_j(t)$ are normalized to the number of variables N in the initial formula, so the definition potential must be modified as follows:

$$V(b, t, \alpha) \equiv -\frac{\gamma'(b, t, \alpha)}{1-t} + b + (1-b) \log(1-b), \quad (4.45)$$

$$\gamma(b, t, \alpha) \equiv \sum_{j=1}^k c_j(t) b^j. \quad (4.46)$$

where the sum over j can be extended to include 1 because $c_1(t) \equiv 0$. V depends on t and α through γ and therefore through the $\{c_j(t)\}$ (which depend on α because of the initial condition). One should be careful not to confuse the time t which appears in these equations with that introduced in the description of the leaf-removal of Section 4.1: t is the fraction of variables appearing in the original formula that have been assigned to obtain the sub-formula, to which the leaf-removal can then be applied.

In equation 4.45 the prime in $\gamma'(b, t, \alpha)$ denotes the partial derivative with respect to b . In the following I shall always denote derivatives with respect to b with primes, and derivatives with respect to t with dots (e.g. $\dot{\gamma}(b, t, \alpha)$). Derivatives with respect to α will be written explicitly.

It is convenient to supplement (4.45) and (4.46) with the generating function of the $\{\rho_j(t)\}$:

$$\phi(b, t, \alpha) = \sum_{j=1}^k \rho_j(t) b^j \quad (4.47)$$

which will play an important role in the following.

4.3.2 General properties of poissonian heuristics

The rate at which clauses are removed from the formula is given by

$$-\sum_{j=1}^k \dot{c}_j(t) = -\dot{\gamma}(1, t, \alpha) = \sum_{j=1}^k \rho_j(t) \quad (4.48)$$

where the “telescopic” terms $(j+1)c_{j+1} - jc_j$ in \dot{c}_j cancel each other. Since at each time step at most one clause is removed from the formula, one must have

$$-\dot{\gamma}(1, t, \alpha) \leq 1. \quad (4.49)$$

This bound is saturated when $\rho_1(t) = 1$ which is the condition for the onset of contradictions.

Moreover, we can multiply (4.41) by j and sum over j to obtain

$$\sum_{j=1}^k j \rho_j(t) = \phi'(1, t, \alpha) = \lim_{\Delta T \rightarrow \infty} \frac{1}{\Delta T} \lim_{N \rightarrow \infty} \sum_{T=Nt}^{Nt+\Delta t-1} \sum_{j=1}^k p_j(\{C_j(T)\}) \leq 1 \quad (4.50)$$

because of the normalization condition (4.37).

More generally, if we denote the average over ΔT which appears in (4.41) and (4.50) with angled brackets $\langle \cdot \rangle$, we have

$$\rho_j(t) = \langle p_j \rangle - \langle p_{j+1} \rangle \quad (4.51)$$

where each $\langle p_j \rangle$ is non-negative and they are normalized so that $\sum_{j=0}^k \langle p_j \rangle = 1$ (because each term in the sum defining the average over ΔT has these properties). Then we have

$$\phi(b, t, \alpha) = \sum_{j=1}^k \rho_j(t) b^j = \sum_{j=1}^k \langle p_j \rangle b^j - \sum_{j=2}^k \langle p_j \rangle b^{j-1} \leq b \sum_{j=1}^k \langle p_j \rangle b^{j-1} \leq b \quad (4.52)$$

since $b \in [0, 1]$. Moreover,

$$\phi'(b, t, \alpha) = \sum_{j=1}^k j \rho_j(t) b^{j-1} = \langle p_1 \rangle + \sum_{j=2}^k b^{j-2} [1 - j(1-b)] \langle p_j \rangle. \quad (4.53)$$

The coefficient in front of $\langle p_j \rangle$ in the terms of the sum is maximum for $b = 1$, independently of j , and is then equal to 1, so that

$$\phi'(b, t, \alpha) \leq \langle p_1 \rangle + \sum_{j=2}^k \langle p_j \rangle = 1 - \langle p_0 \rangle \leq 1. \quad (4.54)$$

These two bounds will be extremely useful in order to characterize the trajectories traced by poissonian heuristics. To do that, for each value of α in the original formula, we can define the three times $t_c(\alpha)$, $t_s(\alpha)$ and $t_q(\alpha)$ at which the reduced sub-formulae cross respectively the clustering transition surface Σ_c , the SAT/UNSAT transition surface Σ_s and the contradiction surface Σ_q defined at the end of Section 4.2. *A priori* we could expect the trajectories to cross each surface more than once, and in this case we shall consider the times of first crossing. By doing this, we ensure that the three functions $t_x(\alpha)$ (where 'x' is 'c', 's' or 'q') are invertible, and we can define $\alpha_x(t)$ as the value of α such that $t_x = t$. On the other hand, it is possible that the trajectory never cross some (or all) of these surfaces, in which case the corresponding $t_x(\alpha)$ will be undetermined.

Since the phase transitions are completely characterized by the potential V , these crossing times will be determined by the conditions (4.32) and (4.33) on $V(b, t, \alpha)$: for given α and t (and therefore for given $\{c_j\}$) we define b^* by (4.31) as the largest solution of the equation $V'(b, t, \alpha) = 0$; then the clustering time $t_c(\alpha)$ will be such that $V''(b^*, t_c, \alpha) = 0$ and the SAT/UNSAT time $t_s(\alpha)$ will be such that $V(b^*, t_s, \alpha) = 0$. As for the contradiction time $t_q(\alpha)$, it is determined by the condition $2c_2(t_q)/(1 - t_q) = 1$.

Let us take the total time derivative of the condition that determines b^* , i.e. $V'(b^*, t, \alpha) = 0$:

$$\frac{d}{dt} V'(b^*, t, \alpha) = V''(b^*, t, \alpha) \frac{db^*}{dt} + \frac{\partial}{\partial t} V'(b^*, t, \alpha) + \frac{\partial}{\partial \alpha} V'(b^*, t, \alpha) \frac{d\alpha}{dt} = 0. \quad (4.55)$$

The term in db^*/dt is present because when t changes, so do the values of $c_j(t)$ and therefore the coefficients in the power series that defines V , and the point where its derivative vanishes moves. In the same manner, if b^* is held fixed, then as t varies the only remaining parameter must vary as well, and this is α , which gives rise to the term in $d\alpha/dt$.

At the clustering transition, $\alpha = \alpha_c(t)$ and $b^* = b_c^*$, the condition $V''(b_c^*, t, \alpha_c(t)) = 0$ is verified, so that the previous equation becomes

$$\frac{d\alpha_c(t)}{dt} = - \frac{\dot{V}'(b_c^*, t, \alpha_c)}{\partial_\alpha V'(b_c^*, t, \alpha_c)} \quad (4.56)$$

where, let me stress it again, $\alpha_c \equiv \alpha_c(t)$ is the value of α such that the trajectory crosses Σ_c at time t , and where the dot denotes a *partial* time derivative.

From the definition (4.45) we have:

$$\dot{V}'(b, t, \alpha) = - \frac{1}{1-t} \left[\dot{\gamma}'(b, t, \alpha) + \frac{\gamma'(b, t, \alpha)}{1-t} \right], \quad (4.57)$$

$$\partial_\alpha V'(b, t, \alpha) = - \frac{1}{1-t} \partial_\alpha \gamma'(b, t, \alpha). \quad (4.58)$$

We can substitute these two expressions into (4.56) to obtain:

$$\frac{d\alpha_c(t)}{dt} = - \frac{\dot{\gamma}'(b, t, \alpha) + \gamma'(b, t, \alpha)/(1-t)}{\partial_\alpha \gamma'(b, t, \alpha)} \Big|_{b=b_c^*(t), \alpha=\alpha_c(t)}. \quad (4.59)$$

From the equations of motion of the heuristic (4.40) we obtain the following equation for $\dot{\gamma}$:

$$\begin{aligned}\dot{\gamma}(b, t, \alpha) &= \sum_{j=1}^k \frac{dc_j}{dt} b^j = \sum_{j=1}^k \left[\frac{(j+1)c_{j+1} - jc_j}{1-t} - \rho_j \right] b^j \\ &= \frac{1-b}{1-t} \gamma'(b, t, \alpha) - \phi(b, t, \alpha).\end{aligned}\quad (4.60)$$

Differentiating it with respect to b we have:

$$\dot{\gamma}'(b, t, \alpha) = -\frac{1}{1-t} \gamma'(b, t, \alpha) + \frac{1-b}{1-t} \gamma''(b, t, \alpha) - \phi'(b, t, \alpha). \quad (4.61)$$

For $b = b_c^*(t)$ we shall have $V' = V'' = 0$, and since

$$V''(b, t, \alpha) = -\frac{1}{1-t} \gamma''(b, t, \alpha) + \frac{1}{1-b} \quad (4.62)$$

we get

$$\left[\frac{1-b}{1-t} \gamma''(b, t, \alpha) = 1 \right]_{b=b_c^*(t), \alpha=\alpha_c(t)} \quad (4.63)$$

so that the numerator in (4.59) becomes $1 - \phi'(b_c^*, t, \alpha_c)$ and we obtain:

$$\frac{d\alpha_c(t)}{dt} = - \left. \frac{1 - \phi'(b, t, \alpha)}{\partial_\alpha \gamma'(b, t, \alpha)} \right|_{b=b_c^*(t), \alpha=\alpha_c(t)}. \quad (4.64)$$

This is where the bounds (4.52) and (4.54) are important (actually, it's only the second of the two which is used here): since $\phi'(b, t, \alpha) \leq 1$ for any b, t, α , the numerator is surely positive or null. Moreover, the denominator is positive at $t = 0$, when $\gamma'(b, 0, \alpha) = \alpha k b^{k-1}$ independently of the heuristic. We then have two cases:

Case 1 The denominator remains positive at all times, in which case $d\alpha_c(t)/dt$ is always negative and $\alpha_c(t)$ is a decreasing function of t , which implies that $t_c(\alpha)$ is a decreasing function of α ;

Case 2 If $\partial_\alpha \gamma'(b, t, \alpha)$ vanishes for some value of t (for a given α), the denominator in (4.64) vanishes. Then $\partial_\alpha t_c(\alpha) = 0$ and $t_c(\alpha)$ has either an extremum or an inflection point. After that, the curve will continue (with decreasing values of α). The curve of $t_c(\alpha)$ cannot reach the axis $\alpha = 0$ (because for $\alpha = 0$ the formula is surely unclustered, and there is no t_c), and neither can it reach the $t = 0$ axis (because at $t = 0$ we have a pure k -XORSAT formula, and we know that it has a unique clustering transition), so it will end at some terminal point.

In both cases, $t_c(\alpha)$ is a single valued function of α . It is the numerator of (4.64), not the denominator, which should change sign in order for $t_c(\alpha)$ to take multiple values. But this cannot happen because of (4.54). An illustration of the possible shapes of the curves for $t_x(\alpha)$ is given in Figure 4.3.

Notice that, even though we considered initially the possibility that the trajectory cross several times Σ_c , and defined t_c as the time of the *first* crossing, the argument I just exposed shows that there can be at most one crossing. We shall see that this fact has profound implications for the performance of poissonian heuristics. Before doing that, however, let me derive an analogous argument for $t_s(\alpha)$.

We start by taking the total time derivative of the potential,

$$\frac{d}{dt} V(b, t, \alpha) = V'(b, t, \alpha) \frac{db}{dt} + \dot{V}(b, t, \alpha) + \partial_\alpha V(b, t, \alpha) \frac{d\alpha}{dt}. \quad (4.65)$$

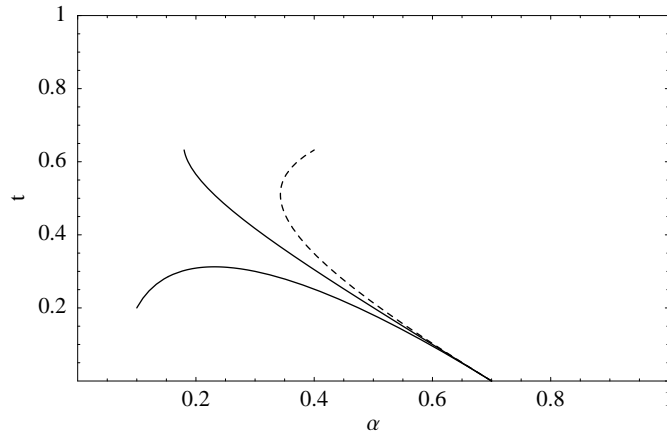


Figure 4.3: Possible shapes for the curves $t_x(\alpha)$ ('x' being 'c' or 's'). t_x is a strictly decreasing function of α if the denominator in (4.64) or (4.72) never vanishes (middle full curve). If instead it does vanish and then changes sign, t_x will develop a maximum and then continue to the left with positive derivative, but it will remain a single-valued function of α (bottom full curve). What cannot occur (top dashed curve) is that $t'_x(\alpha)$ diverges and then changes sign, making t_x a multiple-valued function of α : this would require the numerator in (4.64) or (4.72) to become negative, which cannot occur because of the bounds (4.52) and (4.54). In Section 4.4 I shall prove that actually the curve representing $t_x(\alpha)$ must end at a point where its derivative is infinite, as in the case of the middle full curve.

At the SAT/UNSAT transition, $b = b_s^*(t)$, $\alpha = \alpha_s(t)$ and $V = V' = 0$ from (4.31) and (4.33), so that we obtain:

$$0 = \left[\dot{V}(b, t, \alpha) + \partial_\alpha V(b, t, \alpha) \frac{d\alpha}{dt} \right]_{b=b_s^*(t), \alpha=\alpha_s(t)} \quad (4.66)$$

from which

$$\frac{d\alpha_s}{dt} = - \frac{\dot{V}(b, t, \alpha)}{\partial_\alpha V(b, t, \alpha)} \Big|_{b=b_s^*(t), \alpha=\alpha_s(t)}. \quad (4.67)$$

We can now substitute (4.60) in the partial time derivative of the potential (4.45) to obtain:

$$\begin{aligned} \dot{V}(b, t, \alpha) &= -\frac{1}{1-t} \left[\dot{\gamma}(b, t, \alpha) + \frac{\gamma(b, t, \alpha)}{1-t} \right] \\ &= -\frac{1}{1-t} \left[\frac{1-b}{1-t} \gamma'(b, t, \alpha) - \phi(b, t, \alpha) + \frac{\gamma(b, t, \alpha)}{1-t} \right]. \end{aligned} \quad (4.68)$$

At the SAT/UNSAT transition we have

$$V(b_s^*, t, \alpha_s) = 0 \Rightarrow \frac{\gamma(b_s^*, t, \alpha_s)}{1-t} = b_s^* + (1-b_s^*) \log(1-b_s^*) \quad (4.69)$$

$$V'(b_s^*, t, \alpha_s) = 0 \Rightarrow \frac{\gamma'(b_s^*, t, \alpha_s)}{1-t} = -\log(1-b_s^*) \quad (4.70)$$

so that (4.68) reduces to

$$\dot{V}(b_s^*, t, \alpha_s) = -\frac{1}{1-t} [b_s^* - \phi(b_s^*, t, \alpha_s)]. \quad (4.71)$$

By substituting this in the numerator of (4.67) we obtain:

$$\frac{d\alpha_s(t)}{dt} = - \frac{b - \phi(b, t, \alpha)}{\partial_\alpha \gamma(b, t, \alpha)} \Big|_{b=b_s^*(t), \alpha=\alpha_s(t)}. \quad (4.72)$$

The argument now goes as for $\alpha_c(t)$: the bound (4.52) ensures that the numerator is non-negative, and the denominator is positive at $t = 0$, so that $t_s(\alpha)$ must be single valued.

To summarize, in this paragraph I have shown that the trajectories described by poissonian heuristics can cross the clustering transition surface Σ_c and the SAT/UNSAT transition surface Σ_s only once. Moreover, it is clear that if they reach the contradiction surface Σ_q the algorithm stops, and the crossing of Σ_q must also be unique.

4.3.3 Analysis of UC and GUC

In this paragraph I shall give some examples of the results of the previous paragraph based on two poissonian heuristics that are particularly simple to analyze: UC and GUC.

Analysis of UC

The equations of motion for UC are obtained from (4.38) and (4.40):

$$\frac{dc_j}{dt} = \frac{(j+1)c_{j+1} - jc_j}{1-t} \quad (j \geq 2) \quad (4.73)$$

with the initial condition $c_j(0) = \alpha \delta_{j,k}$. The solution is straightforward:

$$c_j^{\text{UC}}(t) = \alpha \binom{k}{j} (1-t)^j t^{k-j} \quad (j \geq 2). \quad (4.74)$$

As usual $\rho_1 = 2c_2^{\text{UC}}/(1-t)$ and for all $j > 1$ the corresponding $\rho_j = 0$. This is a direct consequence of (4.38): the requirement for Unit Propagation is that the above expression of ρ_1 be true, and if there are no unit clauses $p_0 = 1$ and all the other p_j 's are 0.

We can explicitly compute γ , V and ϕ :

$$\gamma^{\text{UC}}(b, t, \alpha) = \sum_{j=2}^k c_j^{\text{UC}}(t) b^j = \alpha [t + b(1-t)]^k - \alpha k(1-t)t^{k-1}b - \alpha t^k, \quad (4.75)$$

$$\begin{aligned} V^{\text{UC}}(b, t, \alpha) &= -\frac{\gamma'(b, t, \alpha)}{1-t} + b + (1-b) \log(1-b) \\ &= \alpha k t^{k-1} - \alpha k [t + b(1-t)]^{k-1} + b + (1-b) \log(1-b), \end{aligned} \quad (4.76)$$

$$\phi^{\text{UC}}(b, t, \alpha) = \sum_{j=1}^k \rho_j(t) b^j = \rho_1(t) b = \frac{2c_2^{\text{UC}}(t)}{1-t} b = \alpha k(k-1)(1-t)t^{k-2}b. \quad (4.77)$$

An example of the potential $V^{\text{UC}}(b, t, \alpha)$ for $k = 3$ is plotted as a function of b for different values of t and for fixed $\alpha = 0.8$ in Figure 4.1.

The times at which the trajectories cross Σ_c and Σ_s are obtained by solving (numerically) for b and t with fixed α the equations $\{(V^{\text{UC}'} = 0) \wedge (V^{\text{UC}''} = 0)\}$ and $\{(V^{\text{UC}'} = 0) \wedge (V^{\text{UC}} = 0)\}$ (respectively). The bounds (4.52) and (4.54) obviously hold, since ϕ^{UC} is simply $\rho_1 b$ and $\rho_1 \leq 1$, with the equal sign on the contradiction surface Σ_q . Moreover, the denominator in (4.64) is $\partial_\alpha \gamma' = k\{[t + b(1-t)]^{k-1} - t^{k-1}\} > 0$, and the denominator in (4.72) is $\partial_\alpha \gamma$ which is also strictly positive. This ensures that $t_c(\alpha)$ and $t_s(\alpha)$ are strictly decreasing functions of α . The time at which contradictions are generated with probability 1 is obtained by solving $2c_2^{\text{UC}}(t)/(1-t) = 1$ for t at fixed α . The plots of $t_c(\alpha)$, $t_s(\alpha)$ and $t_q(\alpha)$ are shown in Figure 4.4.

The largest value of α for which the algorithm finds a solution with finite probability (which I shall denote $\alpha_h^{\text{UC}}(k)$, the ‘h’ standing for ‘heuristic’) is the smallest value of α for which the trajectory

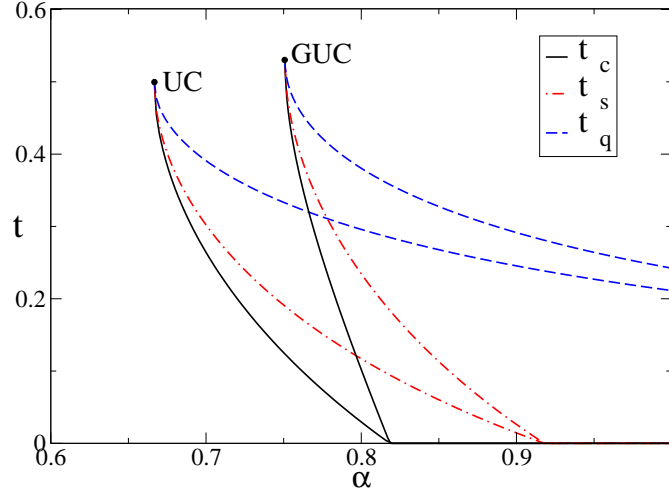


Figure 4.4: Times of crossing of Σ_c , Σ_s and Σ_q for $k = 3$ for UC and GUC. For $\alpha = \alpha_c \simeq 0.818$ the initial formula is at the clustering transition and $t_c = 0$ for both heuristics. The same happens with the SAT/UNSAT transition at $\alpha = \alpha_s \simeq 0.918$. As expected, $t_c(\alpha)$ and $t_s(\alpha)$ are single-valued. The fact that they are strictly decreasing means that for UC and GUC the denominators of (4.64) and (4.72) never change sign.

crosses the SAT/UNSAT transition surface Σ_s . Alternatively, it can be computed as the smallest value of α for which the equation $2c_2^{\text{UC}}(t)/(1-t) = 1$ has a solution, which was done in Section 2.4:

$$\alpha_h^{\text{UC}}(k) = \frac{1}{k} \left(\frac{k-1}{k-2} \right)^{k-2}. \quad (4.78)$$

For $k = 3$ this is equal to $2/3$ and for large k it goes as $e/k + O(k^{-2})$.

Analysis of GUC

The analysis of GUC is slightly more complicated. The analysis of Section 2.4 shows that the equations of motion are

$$\frac{dc_j}{dt} = \frac{(j+1)c_{j+1} - jc_j}{1-t} - \delta_{j,j^*(t)} \left[\frac{1}{j} - \frac{(j-1)c_j}{1-t} \right] \quad (j \geq j^*(t)) \quad (4.79)$$

where $j^*(t)$ is the smallest value of j such that $c_j(t) > 0$, assuming the initial condition $c_j(0) = \alpha \delta_{j,k}$. The interpretation of these equations is that GUC always assigns a variable appearing in the shortest clause (or possibly clauses) in the formula. As long as $j^*c_{j^*}/(1-t) \leq 1/(j^*-1)$ the rate at which clauses of length j^*-1 are generated is small enough that they can be removed, and the density of clauses of length j^*-1 remains 0; when this bound is violated, an extensive number of clauses of length j^*-1 accumulates, and c_{j^*-1} becomes positive. I shall call $t^*(j)$ the time at which $c_j(t)$ becomes positive. When this happens, the value of j^* is decreased by 1. The equations (4.79) therefore hold for $j \geq j^*(t)$, while $c_j(t) \equiv 0$ for all $j < j^*(t)$.

Even though it is in principle possible to solve (4.79) exactly for any finite k , the solution becomes more and more complicated as k increases, since it involves matching the solutions of different differential equations at $k-2$ points (at least for α large enough that j^* reaches 2). I shall only give the

example of $k = 3$, for which one obtains:

$$c_3^{\text{GUC}}(t) = \alpha(1-t)^3, \quad (4.80)$$

$$c_2^{\text{GUC}}(t) = \frac{1}{2}(1-t) \{3\alpha [1 - (1-t)^2] + \log(1-t)\}. \quad (4.81)$$

Notice that from (4.79) it is clear that the $\rho_j(t)$ are all 0 except for two of them:

$$\rho_{j^*} = \frac{1}{j^*} - \frac{(j^* - 1)c_{j^*}}{1-t}, \quad (4.82)$$

$$\rho_{j^*-1} = \frac{j^*c_{j^*}}{1-t}. \quad (4.83)$$

For a fixed value of $j^* = \bar{j}$, t varies between $t^*(\bar{j})$ and $t^*(\bar{j} - 1)$, and during this interval of time, $\bar{j}c_{\bar{j}}(t)/(1-t)$ varies between 0 and $1/(\bar{j} - 1)$, so that we have:

$$\frac{1}{j^*(t)} \leq \rho_{j^*(t)} + \rho_{j^*(t)-1} \leq \frac{1}{j^*(t) - 1}. \quad (4.84)$$

It is easy to see that the bound expressed in (4.52) is respected (actually the previous inequality is even more stringent) and that the bound in (4.50) is respected but saturated: $\sum_j j\rho_j(t) = 1$ at all times.

The crossing times of Σ_c , Σ_s and Σ_q are computed solving numerically the equations obtained from the conditions (4.31), (4.32) and (4.33), as for UC. The results are shown for $k = 3$ in Figure 4.4.

The largest value of α for which GUC succeeds with positive probability in finding a solution, $\alpha_h^{\text{GUC}}(k)$, can be found by looking for the value of α for which $\max_{t \in [0,1]} 2c_2^{\text{GUC}}/(1-t) = 1$. For $k = 3$ this gives the equation $6\alpha - \log(6\alpha) = 3$, so that $\alpha_h^{\text{GUC}}(3) \simeq 0.750874$. Notice that this is larger than $\alpha_h^{\text{UC}}(3)$, as could be expected.

4.4 Bounds on the values of α for which poissonian heuristics can succeed

I shall now discuss how the results of the previous Section on the general properties of poissonian heuristics are related to the phase diagram of k -XORSAT, and in particular what consequences this relation has on the performance of poissonian heuristics in the various phases.

At the end of Section 4.2 I have shown that the surfaces Σ_c and Σ_s intersect each other (I called the intersection *critical surface* Σ_k) and that Σ_c , Σ_s and Σ_k are tangent to each other and to the contradiction surface Σ_q . This is a property of the phase diagram of k -XORSAT which has nothing to do with specific DPLL heuristics. However, a continuity argument based on the fact that the trajectories generated by poissonian heuristics can cross the surfaces Σ_c and Σ_s at most once confirms it. The argument goes as follow.

For any heuristic of the poissonian class, there is a threshold $\alpha_h(k)$ below which the heuristic finds a solution with positive probability and above which this probability vanishes. The heuristic fails with probability 1 if the (average) trajectory intersects the contradiction surface Σ_q . Since for $\alpha < \alpha_h$ the trajectory must not intersect Σ_q while for $\alpha > \alpha_h$ it must, by continuity (of the trajectory and its derivatives, and of Σ_q and its derivatives) this implies that the trajectory corresponding to α_h must be *tangent* to Σ_q .

In the same manner, since the trajectories can cross Σ_s at most once, if a trajectory enters the UNSATphase, it cannot escape from it, and the algorithm must fail. This means that for $\alpha < \alpha_h$ the

trajectories must not cross Σ_s , while for $\alpha > \alpha_h$ they must. As before, by continuity this implies that the trajectory corresponding to α_h must be tangent to Σ_s . The same argument can be made to show that it is also tangent to Σ_c .

Finally, since Σ_c , Σ_s and Σ_q intersect on the critical surface Σ_k and the trajectory corresponding to α_h must be tangent to all of them, without crossing any of them, this means that the trajectory must be tangent to each of them *on* the critical surface Σ_k . Therefore, Σ_c , Σ_s and Σ_q are tangent to each other on Σ_k .

Indeed, it is very simple to see that this argument is correct. The point of a trajectory generated by a poissonian heuristic which is closer to the contradiction surface Σ_q will verify the stationarity condition

$$\frac{d}{dt} \frac{2c_2(t)}{1-t} = \frac{2\dot{c}_2(t)}{1-t} + \frac{2c_2(t)}{(1-t)^2} = 0 \quad (4.85)$$

which, together with the equations of motion (4.40) gives

$$\frac{dc_2(t)}{dt} = \frac{3c_3(t) - 2c_2(t)}{1-t} - \rho_2(t) = -\frac{c_2(t)}{1-t}. \quad (4.86)$$

The critical trajectory (i.e. the trajectory corresponding to α_h) will be such that the value of $2c_2(t)/(1-t)$ at the maximum is 1. When this happens, $\rho_1(t) = 1$ so we must have $\rho_2(t) = 0$ (the heuristic only performs Unit Propagations), and we obtain

$$\frac{3c_3(t)}{1-t} = \frac{1}{2} \quad (4.87)$$

which, together with $2c_2(t)/(1-t) = 1$ is the equation of the critical surface Σ_k given in (4.36). As $2c_2(t)/(1-t)$ is maximum in the point of intersection, the trajectory must be tangent to it.

This has a direct implication for the shape of the curves representing $t_c(\alpha)$, $t_s(\alpha)$ and $t_q(\alpha)$: since each of these curves ends for the value of (α, t) that corresponds to the point where the trajectory is tangent to Σ_k , the three curves must end in the same point (which I shall call *critical point*) in the (α, t) plane, and they must be tangent to each other in the critical point. Since at the critical point the trajectory is on the contradiction surface, so that $\rho_1 = 2c_2/(1-t) = 1$, from (4.72) it is clear that $dt_s/d\alpha$ diverges at the critical point, and since the three curves are tangent, they all have infinite derivative. This is clearly seen in Figure 4.4 for UC and GUC with $k = 3$. The value of α of the critical point is the largest value for which the heuristic succeeds with positive probability, i.e. $\alpha_h(k)$.

We can now derive the main result of this Chapter, which follows in a straightforward manner from the previous discussion. The curve representing $t_c(\alpha)$ starts at the point $(\alpha_c(k), 0)$ and ends at the point $(\alpha_h(k), t_k)$. Moreover, $t_c(\alpha)$ is a single valued function of α , and its derivative is negative at $\alpha = \alpha_c(k)$. This implies that

$$\alpha_h(k) < \alpha_c(k) \quad (4.88)$$

i.e. that poissonian heuristics fail with probability 1 in the clustered phase.

This result is, as far as I know, the first that relates the performance of a class of heuristics for DPLL with the properties of the phase diagram of the optimization problem.

4.5 Optimality of GUC for large k

The result of the previous Section states that no poissonian heuristic for DPLL can succeed with positive probability in the clustered phase, i.e. for $\alpha > \alpha_c(k)$. It is then natural to ask what is the

maximum value of α which can actually be attained, and which heuristic reaches it, that is to say, what is the *optimal* heuristic.

It is clear that the optimal heuristic will be the one which minimizes

$$\Delta\alpha_h \equiv \alpha_c - \alpha_h = \int_0^{t_k} dt' \frac{d\alpha_c(t')}{dt'} = - \int_0^{t_k} dt' \left. \frac{1 - \phi'(b, t', \alpha)}{\partial_\alpha \gamma'(b, t, \alpha)} \right|_{b=b_c^*(t'), \alpha=\alpha_c(t')} \quad (4.89)$$

where I used (4.72) and where t_k is the time coordinate of the critical point in the (α, t) plane, which will depend on the heuristic. Finding the optimal heuristic is a very difficult task: on one hand, the functions $\phi'(b, t, \alpha)$, $\gamma'(b, t, \alpha)$, $b_c^*(t)$ and $\alpha_c(t)$ have a highly non-trivial dependence on the parameters which characterize the heuristic, i.e. the probability functions $\{p_j(C_1, \dots, C_k)\}$; on the other hand, the quantity which must be minimized is an integral, which requires a *functional* optimization.

I shall therefore discuss two more accessible results: first, that for finite k GUC *locally* minimizes the *numerator* of (4.89); and second, that in the limit $k \rightarrow \infty$ GUC indeed is optimal, i.e. $\alpha_h(k) \rightarrow \alpha_c(k)$.

The first statement needs clarification: by locally optimize, I mean that on each point of the trajectory described by GUC, it minimizes the numerator in (4.89). This is a much weaker requirement than optimality, because a different trajectory, which is sub-optimal in some points, might turn out to be much better in some other points, and overall be better than GUC. And of course also because the denominator should be considered as well. However, I think this result is interesting because it sheds some light on why it is impossible for poissonian heuristics to penetrate the clustered phase.

Indeed, from the definition of ϕ , which gives

$$\phi'(b, t, \alpha) = \sum_{j=1}^k j \rho_j(t) b^{j-1}, \quad (4.90)$$

and from the bound

$$\sum_{j=1}^k j \rho_j(t) = \phi'(1, t, \alpha) \leq 1 \quad (4.91)$$

it is clear that ϕ' will be maximized (and hence $1 - \phi'$ will be minimized) by taking “the largest possible ρ_j for the smallest possible j ”. This means that a heuristic which tries to minimize the numerator in the integrand that gives $\Delta\alpha$ should always select the variables to assign in the shortest available clauses, and this is exactly what GUC does.

Moreover, I already noted at the end of Section 4.3 that GUC saturates the bound (4.91). This implies that GUC achieves the largest possible value of $\sum_j \rho_j$, which is the rate at which clauses are eliminated from the formula. Since this only happens through Unit Propagations, it also means that GUC achieves the highest possible rate of Unit Propagations per variables assigned, and therefore minimizes the fraction of variables that are assigned random values. I think this argument makes it at least plausible that GUC is actually the best poissonian heuristic.

A much stronger argument can be made to support the claim that GUC indeed is optimal in the limit $k \rightarrow \infty$. From (4.48) and (4.84) we have, integrating dt :

$$\alpha - \int_0^t \frac{dt'}{j^*(t') - 1} \leq -\gamma(1, t, \alpha) \leq \alpha - \int_0^t \frac{dt'}{j^*(t')}. \quad (4.92)$$

This integral over dt is equal to a sum over the values of j between k and $j^*(t)$,

$$\alpha - \sum_{j=j^*(t)}^k \frac{t^*(j) - t^*(j+1)}{j-1} \leq -\gamma(1, t, \alpha) \leq \alpha - \sum_{j=j^*(t)}^k \frac{t^*(j) - t^*(j+1)}{j} \quad (4.93)$$

since $j^*(t)$ is a step-like function, which is a constant $j^*(t) = \bar{j}$ for $t^*(\bar{j}) \leq t < t^*(\bar{j} - 1)$.

It is reasonable to assume that, in the large k limit,

$$t^*(j) - t^*(j + 1) = \frac{1}{k} + o(k^{-1}) \quad (4.94)$$

for most values of j , i.e. for j such that $0 < j/k < 1$. This assumption is well supported by numerical data for k in the range 2^8 to 2^{16} , as we shall see later.

Under this assumption, we obtain

$$-\gamma(1, t, \alpha) = \alpha - \frac{1}{k} \sum_{j=j^*(t)}^k \frac{1}{j}. \quad (4.95)$$

In order for the algorithm to generate a contradiction with probability 1, we must have $2c_2/(1-t) \geq 1$, and to have $c_2 > 0$, $j^*(t)$ must reach 2. So if j^* always remains larger than 2, the algorithm must have a finite probability to succeed. If it does indeed succeed, it stops when $\gamma(1, t, \alpha) = 0$, since $\gamma(1, t, \alpha)$ is the number of clauses in the formula at time t . The smallest value of α for which the algorithm fails with probability 1 is therefore such that

$$0 = \alpha_h^{\text{GUC}}(k) - \frac{1}{k} \sum_{j=2}^k \frac{1}{j} = \alpha_h^{\text{GUC}}(k) - \frac{\log k + O(1)}{k} \quad (\text{for } k \rightarrow \infty) \quad (4.96)$$

where the term $O(1)$ in the numerator comes from the fact that it is possible that for a number of terms of order $o(k)$ the asymptotic expansion (4.94) doesn't hold. We obtain:

$$\alpha_h^{\text{GUC}}(k) = \frac{\log k}{k} + O(k^{-1}) \quad (\text{for } k \rightarrow \infty). \quad (4.97)$$

This is the same scaling that is found for $\alpha_c(k)$ (see Table 3.1), so that to the leading order in k

$$\alpha_h^{\text{GUC}}(k) \sim \alpha_c(k) \quad (\text{for } k \rightarrow \infty). \quad (4.98)$$

Let us now turn to the assumption (4.94). In order to verify it, we have performed a series of numerical simulations, in which the equations of motion of GUC are integrated by finite differences, for values of k equal to the powers of two between 2^8 and 2^{16} . A finite-size scaling (with respect to k) of the results, shown in Figure 4.5, is consistent with the scaling

$$k[t^*(j) - t^*(j + 1)] = 1 + k^\nu \times f(j/k) \quad (4.99)$$

where $f(x)$ is a function independent on k and which goes as $x^{-\mu}$ for $x \rightarrow 0$. The values of μ and ν are found to be both equal to $1/2$. Integrating the scaling form (4.99) with $\mu = \nu = 1/2$ one obtains that the first correction to the leading term $\log k/k$ in $\alpha_h^{\text{GUC}}(k)$ is of order $1/k$, in agreement with the numerical estimates of $\alpha_h^{\text{GUC}}(k)$ which give $\alpha_h^{\text{GUC}}(k) \simeq \log k/k + 2.15/k$.

I believe that the above numerical results make a strong case supporting the assumption (4.94), and therefore the optimality of GUC.

4.6 Conclusions and perspectives

In this Chapter, I have discussed some very general bounds on the performance of poissonian heuristics for DPLL for the solution of k -XORSAT formulæ and for that of its NP-complete extensions, called

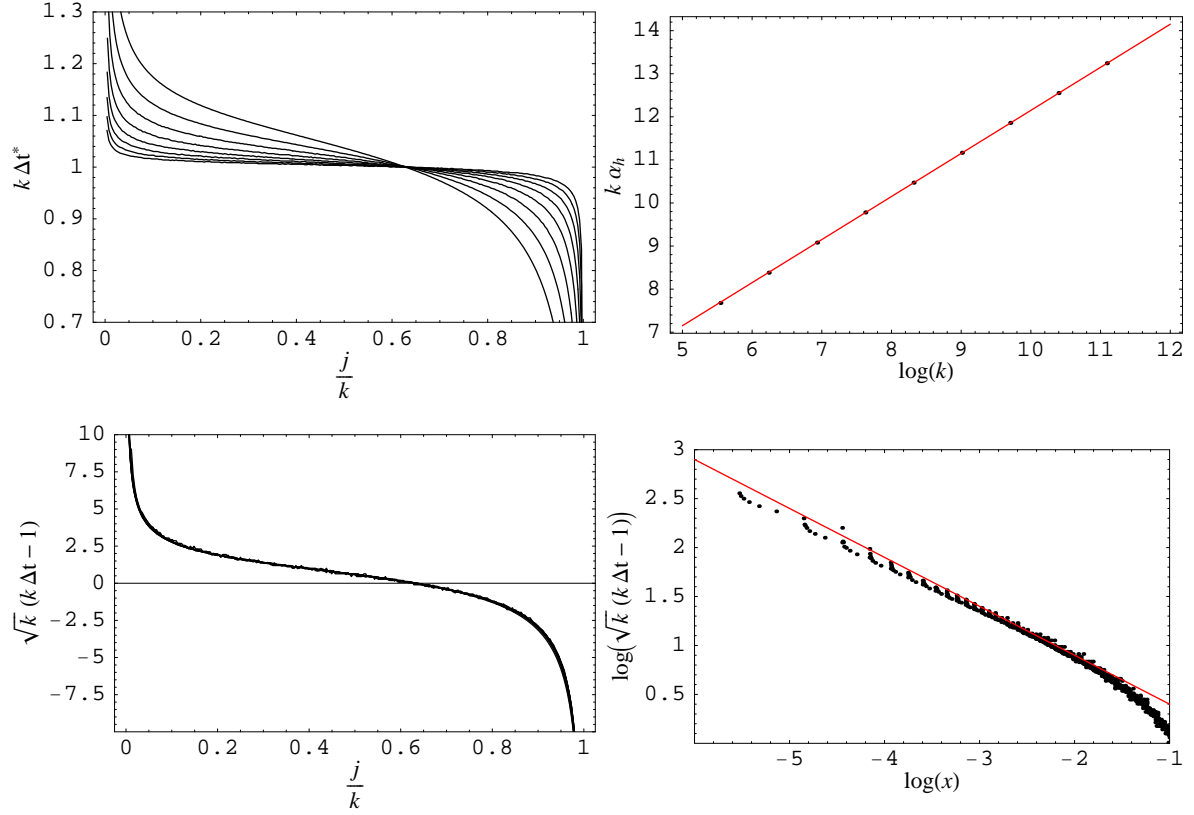


Figure 4.5: Finite size scaling results for GUC at large k . *Top Left* Each curve shows the values of $k[t^*(j) - t^*(j+1)]$ as a function of j/k for $k = 2^8, 2^9, \dots, 2^{16}$ (from the farthest to the closest curve to 1), and was obtained by integrating the equations of motion (4.79) by finite differences. For each k , the value of α used is $\alpha_h^{\text{GUC}}(k)$, determined as the value of α for which the maximum reached by $2c_2(t)/(1-t)$ is 1. *Top Right* Data points of $\alpha_h^{\text{GUC}}(k)$ versus $\log k/k + 2.15/k$ (red line). *Bottom left* The same data as above, plotted as $\{k \times [t^*(j) - t^*(j+1)]\} \times k^{1/2}$. The curves “collapse”, showing $f(x)$ and confirming the value of $\nu = 1/2$. *Bottom right* By plotting the same curves on logarithmic scale it is easily seen that for x close to 0 $f(x) \simeq x^{-\mu}$ with $\mu = 1/2$, corresponding to the slope of the red line.

(d, k) -UE-CSP. In particular, I have proved that such heuristics generate contradictions (i.e. fail) with probability 1 in the clustered phase of the problem.

A point of caution should be placed in the interpretation of this result: it is a very peculiar feature of k -XORSAT that the clustering and freezing transitions coincide. What is found in general in other problems is that the clustering transition, where solutions form an exponential number of connected clusters that are well separated, and the freezing transition, where some variables take a constant value in all the solutions of a given cluster, are distinct. It is well known that in problems where these thresholds are distinct, it is the freezing transition that corresponds to the onset of hardness for known local algorithms. It can be argued that in k -XORSAT too, what causes DPLL poissonian heuristics to fail, is the strong correlations between variables that are present in the frozen phase, rather than the separation of the clusters. In view of this, it would be very interesting to understand what similar bounds could be obtained in problems where the two thresholds are distinct, and notably in k -SAT.

Another interesting question concerns the extension to more general, *non*-poissonian heuristics. In this regard, I have obtained some partial results that seem promising, even though a general theory is still far. More specifically, I have been able to solve the leaf removal equations for the case in which the mixed system to which it is applied is not poissonian, but instead is characterized by some arbitrary distribution of the number of occurrences. However, due to the complicated structure of the solution, it has resulted impossible so far to characterize the phase transitions in terms of a potential, which then would allow to derive some general properties of the trajectories, and possibly some bounds on the values of α for which solutions can be found. Some further work in this direction seems worth undertaking.

Chapter 5

Characterization of the solutions of k -SAT at large α

In this Chapter I shall discuss the properties of the solutions of random k -SAT at large α . This might seem oxymoronic, since at large α random k -SAT formulæ are UNSAT with probability 1. The idea is precisely to restrict the formulæ that are considered to those that, for a given large α , are SAT, then to form an ensemble of these formulæ with uniform weight, and study the properties of their solutions.

Apart from the intrinsic interest of the question, i.e. studying the properties of this particular ensemble of k -SAT formulæ, this problem is relevant because of some recent results by Feige and collaborators [74, 75]: for the first time (as far as I know), they have been able to relate the average case complexity of a satisfiability problem with the worst case complexity of another class of problems, thus bridging the gap between complexity theory and results derived from statistical mechanics methods.

Feige’s result can be summarized as follows: under the assumption that there is no polynomial-time algorithm capable of recognizing *every* SAT instance (and *most* UNSAT instances) of 3-SAT for arbitrarily large (but bounded in N) values of α , the approximation problem to several optimization problems (including min bisection, dense k -subgraph and max bipartite clique) is hard, i.e. non-polynomial in time in the worst case. The complexity class of the approximation problems considered by Feige was previously not known.

With this motivation, Rémi Monasson, Francesco Zamponi and I have studied in [76] the problem of characterizing the solutions of 3-SAT at large α , with the objective of showing that a simple message-passing procedure is able to contradict a probabilistic version of Feige’s assumption, in which “every” is substituted with “with probability p ”, for any (finite) value of p .

In the following Sections, I shall therefore present more in detail Feige’s result and define the problem (Section 5.1); then I shall present the computation of the free energy of the uniform distribution of satisfiable 3-SAT formulæ, in Section 5.2; in Section 5.3 a similar result is derived from the cavity formalism; then, in Section 5.4 I shall compare the results obtained with those that are valid for a different ensemble of formulæ, which was studied by Feige, and draw their algorithmic implications; I shall then comment, in Section 5.5 on the stability of the RS solution of Sections 5.2 and 5.3; finally, in Section 5.6 I shall present and discuss the conclusions of this work.

5.1 Problem definition and previously established results

I shall now define the problem I want to study, and give a brief overview of Feige's results, concerning on one hand the relation between the average-case complexity of 3-SAT and the worst-case complexity of a class of approximation problems, and on the other hand the properties of a very simple message-passing algorithm, which on a particular ensemble of satisfiable 3-SAT formulæ has interesting properties (in view of the previous complexity result).

5.1.1 Definition of the random ensembles

Let us consider random 3-SAT formulæ \mathcal{F} involving N boolean variables $\{x_1, \dots, x_N\}$ and $M = \alpha N$ clauses, with finite α (as $N \rightarrow \infty$). I shall denote assignments of the N variables as $X \equiv \{x_i | i = 1, \dots, N\} \in \{\text{TRUE}, \text{FALSE}\}^N$. Alternatively, I shall represent them as configurations of N Ising spins $\sigma_i \in \{-1, 1\}$, collectively denoted by $\sigma \equiv \{\sigma_i | i = 1, \dots, N\}$, with $\sigma_i = 1$ corresponding to $x_i = \text{TRUE}$ and -1 to FALSE .

The Uniform Ensemble $\mathcal{P}_{\text{Unif}}[\mathcal{F}]$ is obtained by giving the same weight to each possible formula \mathcal{F} . When $\alpha > \alpha_s(3) \simeq 4.267$, the probability over $\mathcal{P}_{\text{Unif}}[\mathcal{F}]$ that a formula \mathcal{F} is SAT is 0: the overwhelming majority of formulæ are UNSAT. It is therefore interesting to introduce two particular ensembles that include only those formulæ that are SAT:

Satisfiable Ensemble \mathcal{P}_{Sat} is the ensemble of satisfiable formulæ, with uniform weight. This is simply the restriction of $\mathcal{P}_{\text{Unif}}$ to satisfiable formulæ.

Planted Ensemble Given an assignment X , the ensemble $\mathcal{P}_{\text{Plant}}^X[\mathcal{F}]$ of SAT formulæ “planted on X ” is defined as the uniform ensemble of formulæ that admit X as a solution. The Planted Ensemble $\mathcal{P}_{\text{Plant}}[\mathcal{F}]$ is obtained by averaging over X with uniform weight for all possible configurations.

Notice that any satisfiable formula is present in both ensembles, but with different weights, as is easily seen from a simple computation: for each clause involving k literals, there is only one assignment of the corresponding k variables that is not SAT. The number of formulæ $\mathcal{N}_f[X]$ that admit X as a solution is therefore

$$\mathcal{N}_f[X] = \left[\binom{N}{k} (2^k - 1) \right]^M \equiv \mathcal{N}_f \quad (5.1)$$

which is independent on X . The Planted Ensemble is then by definition

$$\mathcal{P}_{\text{Plant}}[\mathcal{F}] = \frac{1}{2^N} \sum_X \frac{\mathbb{I}[\mathcal{F} \text{ is satisfied by } X]}{\mathcal{N}_f[X]} = \frac{\mathcal{N}_s[\mathcal{F}]}{2^N \mathcal{N}_f} \quad (5.2)$$

where $\mathcal{N}_s[\mathcal{F}]$ is the number of solutions admitted by \mathcal{F} . It is then clear that $\mathcal{P}_{\text{Plant}}[\mathcal{F}]$ is not uniform, but proportional to the number of solutions of \mathcal{F} .

As we shall see in the following paragraphs, the two ensembles \mathcal{P}_{Sat} and $\mathcal{P}_{\text{Plant}}$ appear in Feige's results.

5.1.2 Hardness of approximation results

In this paragraph I shall give a very brief (and non-rigorous) overview of a theorem proved by Feige in [74].

Feige considers a class of algorithms that take a 3-SAT formula as an input and have two possible outputs: either SAT or UNSAT. The algorithms in question need not be deterministic: for a given

formula, it is admissible that the output be a random variable, whose distribution will then depend on the formula. Notice that, since there are two incompatible outputs, algorithms of this kind can give a wrong answer. However, we shall consider only *asymmetric* algorithms, i.e. such that if the input formula is SAT then the output is *always* SAT; on the other hand, it is admissible that if the input formula is UNSAT the output be SAT, and we shall only require that the probability of this error be smaller than $1/2$ (or some other finite constant, the actual value of which is unimportant).

He then examines the following

Hypothesis 1 Even when α is an arbitrarily large constant (independent on N), there is no polynomial time algorithm that refutes most Random-3-SAT formulæ and *never* wrongly refutes a satisfiable formula.

This hypothesis states that no algorithm of the class described above can work in polynomial time *on average* for 3-SAT formulæ drawn from the Uniform Ensemble $\mathcal{P}_{\text{Unif}}$. In the statement of this Hypothesis, the crucial word *never* refers both to the choice of the formula and to the random moves of the algorithm. According to the author, no algorithms are known to contradict it. Notice that numerical experiments demonstrate that as α grows beyond the SAT/UNSAT transition threshold, k -SAT becomes more “easy” (i.e. the average running time for refutation decreases). However, all known algorithms remain exponential time, and it is only the prefactor of the exponent which decreases. Therefore this observation does not contradict Hypothesis 1. Also, notice that the fact that α is a constant independent of N is crucial: polynomial time algorithms *are* known for $\alpha \gg N^{1/2}$.

In his paper Feige also considers a weaker form of this hypothesis, which has several advantages. The motivation for it is the following. For large α , not only typical random formulæ are UNSAT, but the number of violated constraints becomes concentrated (relative to the Uniform Ensemble of formulæ) around $M/8$, for *every* assignment. Therefore, the formulæ that are *not typical* include all satisfiable formulæ, and also all the formulæ that admit at least one assignment which violates a number of clauses ϵM with $0 < \epsilon < 1/8$.

Hypothesis 2 For every fixed $\epsilon \geq 0$, even when α is an arbitrarily large constant (independent on N), there is no polynomial time algorithm that on most Random-3-SAT formulæ outputs TYPICAL and *never* outputs TYPICAL on formulæ with $(1 - \epsilon)M$ satisfiable clauses.

In this case the algorithm considered has two possible outputs, TYPICAL and NOT TYPICAL, and again the admissible error is asymmetric. For $\epsilon = 0$ Hypothesis 2 reduces to Hypothesis 1.

Notice that, despite the appearance, Hypothesis 1 implies Hypothesis 2 and therefore Hypothesis 2 is *weaker* than Hypothesis 1. In order to realize it, let me show that if Hypothesis 2 is violated, then Hypothesis 1 is also violated. Indeed, if Hypothesis 2 is violated, an algorithm exists which is able to identify formulæ that have a fraction of satisfiable clauses larger than $1 - 1/8$. In most cases the output of this algorithm will be TYPICAL, meaning that the fraction of satisfiable clauses is $1 - 1/8$; however, if the formula has a fraction of satisfiable clauses larger than $1 - 1/8$, it will be identified as such. Therefore, such an algorithm will output TYPICAL most of the time, but it will never output TYPICAL if the formula is SAT (and therefore has a fraction of satisfiable clauses larger than $1 - 1/8$), thus contradicting Hypothesis 1.

The main result from [74] is the following

Theorem 1 The existence of an algorithm able to approximate in polynomial time the solution to any of the following problems would contradict Hypothesis 2: min bisection, dense k -subgraph, max bipartite clique (all within a constant approximation factor) and 2-catalog (within a factor N^δ where N is the number of edges and $0 < \delta < 1$ some constant).

I shall not define these problems, which are well known in theoretical computer science and of little interest for the following¹. It suffices me to say that their complexity class is not known. If Hypothesis 2 were proved to be true, as a consequence all these problems would be NP-hard, and this would be an interesting new result.

As I already mentioned, this theorem establishes a relation between the *average-case complexity* of 3-SAT at large α and the *worst-case complexity* of some other problems. In this regard, it is a very striking result, and it opens the possibility of applying statistical mechanics methods to complexity theory.

Without any ambition to rigor, let me just sketch the proof of the theorem, which is rather interesting. Let us define a problem P as R-3-SAT-hard if the existence of a polynomial time algorithm to solve P would contradict Hypothesis 2. In particular, a problem is R-3-SAT-hard if it is possible to reduce any instance of 3-SAT to an instance of P to which A can be applied, in such a way as to contradict Hypothesis 2. Then, Feige proves that several other boolean constraint satisfaction problems, and their optimization versions, are R-3-SAT-hard.

More specifically, let us consider a boolean function over three variables, $f : \{\text{TRUE}, \text{FALSE}\}^3 \rightarrow \{\text{TRUE}, \text{FALSE}\}$. The number of such functions is 2^{2^3} , most of which coincide up to renaming or negation of the variables. For each of them, let us define as t the number of possible inputs, out of 2^3 , for which f evaluates to TRUE, and b (for *bias*) the number of possible inputs with an *odd* number of TRUE values and for which f evaluates to TRUE (or, if it is larger, the same quantity with *even* instead of *odd*). Then, there are 13 *distinct* such functions for which $2b > t$, including AND, OR and XOR.

Consider a “ $3f$ -clause” involving 3 literals over N variables and based on any of these 13 functions f , and a random “ $3f$ -formula” made of $M = \alpha N$ such clauses. Feige proves the following

Theorem 2 It is R-3-SAT-hard to distinguish between those random $3f$ -formulae in which a fraction just over $t/8$ of the clauses are satisfied, and those in which this fraction is just below $b/4$ (assuming α is sufficiently large). In particular, this implies that it is R-3-SAT-hard to approximate MAX- $3f$ within a constant factor better than $t/2b$.

This theorem is very interesting in itself: it is here that the link between the complexity of a decision problem (namely R-3-SAT) and that of an approximation problem is established (even though, only for the average case). The proof of Theorem 2 is straightforward but complicated, and I shall omit it.

Feige then proves the following

Proposition For every $\epsilon > 0$, there is an α_ϵ such that for any $\alpha > \alpha_\epsilon$ and N large enough, with probability 1 the following holds: *every* set of $(1/8 + \epsilon)M$ clauses in a R-3-SAT formula with $M = \alpha N$ clauses contains at least $N + 1$ distinct literals.

The crucial point, which will allow to establish a link between average-case and worst-case complexity, is that the proposition holds, with probability 1 over the choice of the 3-SAT formula, for *every* set of $(1/8 + \epsilon)M$ clauses of a given formula. The proof of this proposition is rather simple: given N variables, corresponding to $2N$ literals, let us select a set S containing N literals. The probability that a random clause contains no literal from S is $(1/2)^3$, and the probability that m clauses out of M contain no literals from S is

$$P_S(m) = \binom{M}{m} \left(\frac{1}{2}\right)^{3m} \left[1 - \left(\frac{1}{2}\right)^3\right]^{M-m} \quad (5.3)$$

¹A definition is given in [74]

which, for large M and $m = \mu M$, is asymptotically

$$P_S(m) \sim \exp \{M \log 2 [-\mu \log_2 \mu - (1 - \mu) \log_2 (1 - \mu) - 3\mu + (1 - \mu) \log_2 (7/8)]\} \equiv e^{M\phi(\mu)}. \quad (5.4)$$

This probability is maximum for $\mu = 1/8$, and verifies the large deviations relation

$$\mathbb{P} \left[\mu = \frac{1}{8} + \epsilon \right] \sim \exp \left[\alpha N \phi''(1/8) \frac{\epsilon^2}{2} \right] \quad (5.5)$$

with $\phi''(1/8) = -64/7$.

Therefore, for any given $\epsilon > 0$, provided $\alpha > -3 \times 2 / [\phi''(1/8)\epsilon^2]$, we shall have

$$\mathbb{P} \left[\mu = \frac{1}{8} + \epsilon \right] < 2^{-3N} \Rightarrow \mathbb{P} \left[\mu < \frac{1}{8} + \epsilon \right] > 1 - 2^{-3N}. \quad (5.6)$$

More explicitly:

$$\mathbb{P} [\text{at least } (1/8 + \epsilon)M \text{ clauses out of } M \text{ contain no literal from } S] < 2^{-3N}. \quad (5.7)$$

We can now use Boole's inequality,

$$\mathbb{P} \left[\bigcup_i A_i \right] \leq \sum_i \mathbb{P}[A_i] \quad (5.8)$$

and write, for all the possible subsets S of N literals out of $2N$,

$$\mathbb{P} \left[\text{at least } (1/8 + \epsilon)M \text{ clauses out of } M \text{ contain no literal from } \bigcup S \right] < \sum_S 2^{-3N} \quad (5.9)$$

$$\Leftrightarrow \mathbb{P} [\text{at least } (1/8 + \epsilon)M \text{ clauses out of } M \text{ contain no literal from *any* set of } N \text{ literals}] < 2^{-N} \quad (5.10)$$

since the number of possible sets S is less than 2^{2N} . This statement is equivalent to the one in the Proposition: every set of $(1/8 + \epsilon)M$ clauses contains at least $N + 1$ literals, with probability 1 over the choice of the formula from which the clauses are taken.

The proof of Theorem 1 then proceeds as follows, for each of the graph-based problems P listed in the enunciate. Some 3AND-formula \mathcal{F} with $M = \alpha N$ clauses in N variables is mapped to a graph \mathcal{G} by a suitable construction. The actual constructions vary with the specific problem P and I shall omit them. Let us make the case of min bisection for concreteness. The Proposition is used to prove that if \mathcal{F} has at most $(1/8 + \epsilon)M$ satisfiable clauses, then the corresponding \mathcal{G} has a cut of width at least $(1 - \epsilon)M$; while if \mathcal{F} has at least $(1/4 - \epsilon)M$ satisfiable clauses, then the corresponding \mathcal{G} has a cut of width $3(1/4 + \epsilon)M$. This means that if it is possible to approximate min bisection on *every* instance within a factor $3/4$, then it is possible to compute the approximate bisection, and from the approximate value it will be possible to distinguish the two cases (i.e. of *typical* 3AND-formulæ with $(1/8 + \epsilon)M$ satisfiable clauses vs. *non-typical* 3AND-formulæ with $(1/4 - \epsilon)M$ satisfiable clauses). Because of Theorem 2, this contradicts Hypothesis 2, and thus proves Theorem 1.

5.1.3 Performance of Warning Propagation on the planted distribution

The problem of establishing or refuting Hypothesis 1 and/or Hypothesis 2 was tackled by Feige and collaborators in [75]. That paper makes a step forwards in the direction of refutation, but does not achieve to prove it in general.

The authors consider a simple message passing procedure, called Warning Propagation (WP). Given a 3-SAT formula \mathcal{F} and the factor graph \mathcal{G} representing it, two kinds of messages are defined for each edge in \mathcal{G} , i.e. for each pair (\mathcal{C}_a, x_i) where \mathcal{C}_a is a clause and x_i a variable appearing in it: *clause-to-variable messages* $u_{a \rightarrow i}$ are binary variables equal to 0 or 1; *variable-to-clause messages* $h_{i \rightarrow a}$ are integer variables (positive, negative or null). The following update rule is defined:

$$\begin{cases} h_{i \rightarrow a} = \sum_{b \in \partial_+ i \setminus a} u_{b \rightarrow i} - \sum_{b \in \partial_- i \setminus a} u_{b \rightarrow i}, \\ u_{a \rightarrow i} = \prod_{j \in \partial a \setminus i} \mathbb{I}[h_{j \rightarrow a} < 0] \end{cases} \quad (5.11)$$

where ∂a is the set of variables appearing in clause \mathcal{C}_a , the back-slash denotes privation, $\partial_+ i$ is the set of clauses in which variable x_i appears non-negated, and $\partial_- i$ is the set of clauses in which it appears negated.

WP is defined as the following algorithm, taking a 3-SAT formula \mathcal{F} as input and returning a *partial* assignment X as output:

procedure WARNING PROPAGATION(\mathcal{F})

Construct the factor graph \mathcal{G} representing \mathcal{F}

Randomly initialize the clause-to-variable messages $\{u_{a \rightarrow i}\}$ to 0 or 1

repeat

Randomly order the edges of \mathcal{G}

Update the messages $h_{i \rightarrow a}$ and $u_{a \rightarrow i}$ in the selected order according to the rule (5.11)

until No message changes in the update

Compute a partial assignment X based on $\{h_{i \rightarrow a}\}$:

if $\sum_{a \in \partial_+ i} h_{i \rightarrow a} - \sum_{a \in \partial_- i} h_{i \rightarrow a} > 0$ **then**

$x_i = \text{TRUE}$

else if $\sum_{a \in \partial_+ i} h_{i \rightarrow a} - \sum_{a \in \partial_- i} h_{i \rightarrow a} < 0$ **then**

$x_i = \text{FALSE}$

else

x_i is unassigned

end if

Return X

end procedure

Notice that some variables in X will be unassigned at the end of WP.

The main result proved in [75] is the following

Theorem 2 For any assignment Y and any formula \mathcal{F} from the ensemble $\mathcal{P}_{\text{Plant}}^Y[\mathcal{F}]$ planted on Y with large enough α (but constant in N), the following is true with probability $1 - e^{-O(\alpha)}$ over the choice of the formula and the random moves of WP:

1. WP(\mathcal{F}) converges after at most $O(\log N)$ iterations
2. The fraction of variables assigned in X is $1 - e^{-O(\alpha)}$, and for each of them $x_i = y_i$ (the value it takes in the planted assignment Y)
3. The formula obtained by simplifying \mathcal{F} with the values assigned in X can be satisfied in time $O(N)$

5.1.4 Discussion of the known results and problem definition

Theorem 2 establishes that WP has some properties of the algorithm described in Hypothesis 1, but with some important differences, as I shall discuss in this paragraph.

First, WP is a constructive algorithm, but it is not complete: it is possible that it never converges (i.e. that the loop goes on for ever); however, if it does converge, it provides an assignment which can be easily checked. One can set a fixed maximum number of iterations \mathcal{N}_i and stop the execution if it is reached; the output will then be UNSAT, and this will possibly be wrong. If, on the contrary, an assignment is returned (and it is checked to be satisfying), the output will be SAT, and this will surely be true.

Therefore WP is an asymmetric algorithm, which never outputs SAT to an UNSAT formula, but which sometimes outputs UNSAT to a SAT formula. The algorithm described in Hypothesis 1 is different in this regard, as it must *never* return UNSAT to a SAT formula.

Second, the statements in Theorem 2 hold in probability for formulæ drawn from the Planted Ensemble, while in Hypothesis 1 the Uniform Ensemble is considered.

The conclusion which can be drawn is that Theorem 2 refutes the following modified

Hypothesis 1_p Planted Even when α is an arbitrarily large constant (independent on N), there is no polynomial time algorithm that refutes most Random-3-SAT formulæ *from the Planted Ensemble* $\mathcal{P}_{\text{Plant}}$, and outputs SAT *with probability* p on a 3-SAT formula which is satisfiable.

The differences relative to Hypothesis 1 are written in italics in Hypothesis 1_p Planted: the distribution of formulæ is the Planted Ensemble instead of the Uniform one, and satisfiable formulæ are recognized with probability p instead of always.

The question I shall try to answer in the rest of this Chapter is if it is possible to make further progress towards the refutation of Hypothesis 1, and in particular if the convergence of WP can be established for formulæ drawn from the Satisfiable Ensemble \mathcal{P}_{Sat} . This is equivalent to proving it for the Uniform Ensemble, since \mathcal{P}_{Sat} is the restriction of $\mathcal{P}_{\text{Unif}}$ to satisfiable formulæ, and for formulæ that are not SAT it is admissible for the algorithm to give wrong answers (i.e. not to converge).

The main conclusion that we shall reach is that this is indeed true, and that the following

Hypothesis 1_p Even when α is an arbitrarily large constant (independent on N), there is no polynomial time algorithm that refutes most Random-3-SAT formulæ $\mathcal{P}_{\text{Plant}}$, and outputs SAT *with probability* p on a 3-SAT formula which is satisfiable.

is wrong for any $p < 1$. A similarly probabilistic version of Hypothesis 2 will also be refuted.

5.2 Free energy of the uniform distribution of satisfiable formulæ

In this Section, I shall apply the replica formalism for diluted systems described in Paragraph 1.4.3 to a spin glass problem which is equivalent to Random-3-SAT, in order to derive the properties of the formulæ in \mathcal{P}_{Sat} and of their solutions.

The following computation follows the one presented in [77], the main difference being the introduction (in Paragraph 5.2.3) of a “chemical potential” that will permit to select only the formulæ that are satisfiable.

5.2.1 Replicated partition function of k -sat

In this Section, we shall use the representation of an assignment X as a collection of N Ising spins, $\sigma \equiv \{\sigma_1, \dots, \sigma_N\}$. For a given k -SAT formula \mathcal{F} and a given configuration σ we define the energy function

$$E_{\mathcal{F}}(\sigma) = \sum_{i=1}^M \mathbb{I}[\sigma \text{ verifies } \mathcal{C}_i] \quad (5.12)$$

where \mathcal{C}_i is the i^{th} clause in \mathcal{F} . This energy is simply the number clauses in \mathcal{F} that are violated by σ .

The partition function is defined as

$$Z_{\mathcal{F}}(\beta) = \sum_{\sigma} e^{-\beta E_{\mathcal{F}}(\sigma)} = \sum_{\sigma} \prod_{i=1}^M z_i(\sigma) \quad (5.13)$$

where $z_i(\sigma) \equiv \exp\{-\beta \mathbb{I}[\sigma \text{ verifies } \mathcal{C}_i]\}$. The average of the replicated partition function over the choice of the formula from the Uniform Ensemble $\mathcal{P}_{\text{unif}}$, which I shall denote by an overline, is

$$\overline{Z(\beta)^n} \equiv \sum_{\mathcal{F}} \mathcal{P}_{\text{Unif}}[\mathcal{F}] Z_{\mathcal{F}}(\beta)^n = \overline{\sum_{\sigma^1, \dots, \sigma^n} \prod_{i=1}^M z_i(\sigma^1) \cdots z_i(\sigma^n)} \quad (5.14)$$

where σ^a is the N -component configuration of replica a .

Since the literals appearing in each clause are extracted independently on the other clauses, the average over the choice of the formula reduces to the average over the literals appearing in a clause, raised to the power M :

$$\overline{Z(\beta)^n} = \sum_{\sigma^1, \dots, \sigma^n} \left[\overline{z(\sigma^1) \cdots z(\sigma^n)} \right]^M. \quad (5.15)$$

Let us consider a term in the sum, corresponding to a given $\sigma \equiv (\sigma^1, \dots, \sigma^n)$. It is the average over the choice of the literals appearing in the clause of a product over the replica index a of a quantity which is 1 if the clause considered is satisfied by replica a and $e^{-\beta}$ otherwise. Let us denote by i_j the index of the j^{th} literal in the clause, and by q_j a variable which is -1 if it is negated and 1 otherwise. We have:

$$\overline{z(\sigma^1) \cdots z(\sigma^n)} = \binom{N}{k}^{-1} \sum_{i_1 < \dots < i_k}^{1, N} \frac{1}{2^k} \sum_{q_1, \dots, q_k}^{\{-1, 1\}} \prod_{a=1}^n \left\{ 1 + (e^{-\beta} - 1) \prod_{j=1}^k \delta(\sigma_{i_j}^a, q_j) \right\} \quad (5.16)$$

where the δ is a Kronecker function. In the following we shall consider the limit $N \rightarrow \infty$, and in view of this we can neglect the constraint of the k indices being different and approximate the binomial with N^k .

The product over a appearing in (5.16) is a function of the replicated configurations $\vec{\sigma}_{i_j}$ at the sites $\{i_1, \dots, i_k\}$. Since we are averaging over the choice of the sites, it is convenient to introduce

$$\rho(\vec{\tau}|\sigma) \equiv \frac{1}{N} \sum_{i=1}^N \prod_{a=1}^n \delta(\tau^a, \sigma_i^a) \quad (5.17)$$

which is the fraction of sites that, for a given σ , are equal to the n -component configuration $\vec{\tau}$. We then have

$$\overline{z(\sigma^1) \cdots z(\sigma^n)} = \sum_{\vec{\tau}_1, \dots, \vec{\tau}_k} \rho(\vec{\tau}_1|\sigma) \cdots \rho(\vec{\tau}_k|\sigma) \mathcal{E}(\vec{\tau}_1, \dots, \vec{\tau}_k) \quad (5.18)$$

where

$$\mathcal{E}(\vec{\tau}_1, \dots, \vec{\tau}_k) \equiv \frac{1}{2^k} \sum_{q_1, \dots, q_k}^{\{-1, 1\}} \prod_{a=1}^n \left\{ 1 + (e^{-\beta} - 1) \prod_{j=1}^k \delta(\tau_j^a, q_j) \right\}. \quad (5.19)$$

The replicated partition function (5.15) is then

$$\overline{Z(\beta)^n} = \sum_{\sigma} \exp \left\{ M \log \sum_{\vec{\tau}_1, \dots, \vec{\tau}_k} \rho(\vec{\tau}_1 | \sigma) \cdots \rho(\vec{\tau}_k | \sigma) \mathcal{E}(\vec{\tau}_1, \dots, \vec{\tau}_k) \right\}. \quad (5.20)$$

We can introduce the function $c(\vec{\tau})$ and multiply the previous expression by the functional integral

$$\int_0^1 \delta c(\cdot) \delta [c(\vec{\tau}) - \rho(\vec{\tau} | \sigma)] = 1 \quad (5.21)$$

(where the integrand is a functional Dirac distribution), to obtain

$$\overline{Z(\beta)^n} = \int_0^1 \delta c(\cdot) \exp \left\{ \alpha N \log \sum_{\vec{\tau}_1, \dots, \vec{\tau}_k} c(\vec{\tau}_1) \cdots c(\vec{\tau}_k) \mathcal{E}(\vec{\tau}_1, \dots, \vec{\tau}_k) \right\} \sum_{\sigma} \delta [c(\cdot) - \rho(\cdot | \sigma)]. \quad (5.22)$$

The sum over σ is the number of n -replicated N -sites configurations such that for any $\vec{\tau}$, the fraction of sites that have a replicated configuration $\vec{\tau}$ is equal to $c(\vec{\tau})$. For each of the possible values of $\vec{\tau}$ one has to choose the $Nc(\vec{\tau})$ sites that will have $\vec{\tau}$ as their replicated configuration, and the number of ways to do it is the multinomial coefficient:

$$\sum_{\sigma} \delta [c(\cdot) - \rho(\cdot | \sigma)] = \frac{N!}{\prod_{\vec{\tau}} [Nc(\vec{\tau})]!} \sim \exp \left[-N \sum_{\vec{\tau}} c(\vec{\tau}) \log c(\vec{\tau}) \right] \quad (5.23)$$

to the leading order as $N \rightarrow \infty$.

The “physical” interpretation of the previous results is the following: the function $c(\cdot)$ is the order parameter of our theory, the term which multiplies N in the exponent of (5.22) is the effective energy expressed in terms of $c(\cdot)$, and (5.23) is the (microcanonical) entropy of $c(\cdot)$. This follows exactly the scheme traced in Paragraph 1.4.3. Notice, moreover, that the “physical” inverse temperature β only appears in the definition of \mathcal{E} , which is the effective interaction strength, and that the parameter which plays the role of the inverse temperature in the effective theory is α .

5.2.2 Free energy and replica symmetric ansatz

We can write (5.22) in terms of an effective free energy density

$$\mathcal{F}[c(\cdot), n, \beta, \alpha] \equiv - \sum_{\vec{\tau}} c(\vec{\tau}) \log c(\vec{\tau}) + \alpha \log \sum_{\vec{\tau}_1, \dots, \vec{\tau}_k} c(\vec{\tau}_1) \cdots c(\vec{\tau}_k) \mathcal{E}(\vec{\tau}_1, \dots, \vec{\tau}_k) \quad (5.24)$$

as the functional integral

$$\overline{Z(\beta)^n} = \int_0^1 \delta c(\cdot) e^{N \mathcal{F}[c(\cdot), n, \beta, \alpha]}, \quad (5.25)$$

which in the thermodynamic limit $N \rightarrow \infty$ can be evaluated by saddle point. The free energy density, defined as

$$f(\beta, \alpha) \equiv \frac{1}{\beta} \lim_{N \rightarrow \infty} \frac{1}{N} \lim_{n \rightarrow 0} \frac{1}{n} \log \overline{Z(\beta)^n} \quad (5.26)$$

is then equal to the limit for $n \rightarrow 0$ of the extremum value of (5.24) over $c(\cdot)$,

$$f(\beta, \alpha) = \lim_{n \rightarrow 0} \frac{1}{\beta n} \operatorname{extremum}_{c(\cdot)} \mathcal{F}[c(\cdot), n, \beta, \alpha]. \quad (5.27)$$

Notice however that, as usual with replica calculations, the order of the two limits $N \rightarrow \infty$ and $n \rightarrow 0$ has been reversed, which has no *a priori* justification.

In order to compute the extremum of the effective free energy, some assumption must be made on the form of the function $c(\cdot)$. The replica symmetric *ansatz* considers only functions that are symmetric in the replica index, of the form:

$$c(\vec{\tau}) = \gamma \left(\sum_{a=1}^n \tau^a \right). \quad (5.28)$$

Under this assumption, a convenient parameterization of the function $\gamma(\cdot)$ is in terms of the auxiliary function $R(h)$

$$c(\vec{\tau}) = \int_{-\infty}^{\infty} dh R(h) \frac{\exp \left\{ \frac{\beta h}{2} \sum_{a=1}^n \tau^a \right\}}{[2 \cosh(\beta h/2)]^n}. \quad (5.29)$$

A few remarks are in order. First, we expect the function $\gamma(\cdot)$ to be even, because in (5.19) we are summing over the values of q_j , and $\overline{Z(\beta)^n}$ is therefore invariant under $\vec{\tau} \rightarrow -\vec{\tau}$. This implies that $R(h)$ must also be even. Second, $c(\vec{\tau})$ is normalized to 1 (it is the fraction of sites that have replicated configuration $\vec{\tau}$), so also $R(h)$ must be normalized,

$$\int_{-\infty}^{\infty} R(h) dh = 1. \quad (5.30)$$

Third, the equation (5.29) defining $R(h)$ is, apart from the factor in the denominator, a Laplace transform, so that $R(h)$ is indeed well defined. Finally, notice that the expression multiplying $R(h)$ in (5.29) is the Gibbs weight of a system of n Ising spins τ^a in a uniform magnetic field h at temperature β . Since the physical interpretation² of $c(\vec{\tau})$ is the fraction of sites having the replicated configuration $\vec{\tau}$, that is to say, the probability that the configuration $\vec{\tau}$ is observed, the interpretation of h is indeed that of a magnetic field acting on the spins, and the interpretation of $R(h)$ is that of the probability distribution of the values of the field h . This observation motivates the introduction of $R(h)$.

5.2.3 Selection of satisfiable formulæ by means of a “chemical potential”

So far, the computation has been performed for any β , and nothing in it ensures that only satisfiable formulæ are considered in the average: the ensemble we are considering is $\mathcal{P}_{\text{Unif}}[\mathcal{F}]$ instead of $\mathcal{P}_{\text{Sat}}[\mathcal{F}]$. I shall now introduce a method which allows to restrict the ensemble to $\mathcal{P}_{\text{Sat}}[\mathcal{F}]$.

General strategy

In all generality, for systems with discrete configurations \mathcal{C} and discrete non-negative energy $E(\mathcal{C}) \in \{E_0, E_1, \dots\}$, the partition function $Z(\beta)$ can be written as

$$Z(\beta) = \sum_{\mathcal{C}} e^{-\beta E(\mathcal{C})} = \sum_i g_i e^{-\beta E_i} \quad (5.31)$$

²This is true for the function $c(\cdot)$ which extremizes the free energy density. The physical interpretation of h and $R(h)$ therefore holds only for the $R(h)$ corresponding to the extremum.

where g_i is the number of configurations with energy E_i . In the presence of disorder, the values of $\{E_i\}$ depend on the sample. The average over disorder of the replicated partition function is then

$$\overline{Z(\beta)^n} = \overline{[g_0 e^{-\beta E_0} + g_1 e^{-\beta E_1} + \dots]^n}. \quad (5.32)$$

In order to compute the free energy of the model, the limit $n \rightarrow 0$ must be taken, and if one is interested in the low temperature behavior of the model, the question of the value of the product $\beta n \equiv \nu$ rises.

Normally, what is needed is the low temperature behavior of the average of the free energy over all values of the disorder, and n must go to 0 *before* sending $\beta \rightarrow \infty$, which corresponds to $\nu = 0$. In our case, however, we would like to select the values of the disorder parameters that minimize the energy of the system, and to restrict the average to these values. Let us see what happens when taking $\beta \rightarrow \infty$ and $n \rightarrow 0$ with finite ν . We can formally develop the multinomial in (5.32) to obtain

$$\begin{aligned} \overline{Z(\beta)^n} &= \overline{g_0^n e^{-\nu E_0}} + n \overline{g_0^{n-1} e^{-\beta(n-1)E_0} [g_1 e^{-\beta E_1} + \dots]} + \dots \\ &= \overline{g_0^n e^{-\nu E_0}} + n \overline{g_0^{n-1} g_1 e^{-\nu E_0} e^{\beta(E_0 - E_1)}} + \dots \end{aligned} \quad (5.33)$$

where each term after the first has a factor $e^{\beta(E_0 - E_i)}$ which makes it vanish, so that only the first term contributes. Since g_0 is independent on n , and $n \rightarrow 0$ we remain with

$$\overline{Z(\beta)^n} \simeq \overline{e^{-\nu E_0}}. \quad (5.34)$$

We see that the consequence of taking $\nu > 0$ is to include in the computation of the replicated partition function, for a given realization of the disorder, only the lowest energy configurations.

The energy E_0 is the *extensive* energy of the ground state of the system for a given realization of the disorder, which can be regarded as a random variable over the distribution of disorder. Let us denote by $\omega(\epsilon)$ the large deviations function of the distribution of the energy density $\epsilon = E_0/N$ of the ground state, i.e.

$$\mathbb{P}[E_0 = N\epsilon] = e^{N\omega(\epsilon) + o(N)}. \quad (5.35)$$

It is reasonable to expect that $\omega(\epsilon)$ will be a negative convex function (i.e. $\omega''(\epsilon) < 0$), vanishing in its maximum. Let's assume that this is the case. For large N we shall have

$$\overline{Z(\beta)^n} \simeq \int d\epsilon e^{N[\omega(\epsilon) - \nu\epsilon]} \simeq e^{N\varphi(\nu)} \quad (5.36)$$

where

$$\varphi(\nu) \equiv \max_{\epsilon} [\omega(\epsilon) - \nu\epsilon] \quad (5.37)$$

is the Legendre transform of $\omega(\epsilon)$, provided the convexity assumption on ω holds (which can be verified *a posteriori*).

The integral in (5.36) will be dominated by the contribution from the value of ϵ which maximizes the exponent, which is given by

$$\epsilon_0(\nu) = -\partial_{\nu}\varphi(\nu). \quad (5.38)$$

The partition function computed in (5.36) is therefore averaged only on those values of disorder that give a ground state energy equal to $N\epsilon_0(\nu)$: the parameter ν allows to restrict the distribution of the disorder to some subset with a well defined ground state energy. In this regard, it plays a role similar to a chemical potential in thermodynamics.

Let us now turn to the application of this program to compute the replica symmetric free energy of k -SAT over the Satisfiable ensemble $\mathcal{P}_{\text{Sat}}[\mathcal{F}]$. The strategy will be to substitute the replica symmetric

ansatz (5.29) for $c(\cdot)$ in the free energy (5.24), and take the limits $\beta \rightarrow \infty$ and $n \rightarrow 0$ with finite $\nu = \beta n$, to obtain a free energy functional depending on ν , analogous to $\varphi(\nu)$ in (5.36), and which will have a functional dependence on $R(h)$. Then to derive the saddle point equation corresponding to (5.27) and which will determine $R(h)$, and solve them for generic ν . We shall compute the average ground state energy as a function of ν , as in (5.38), and find the value of ν corresponding to zero energy, which will select satisfiable formulæ. The equilibrium distribution of fields $R(h)$ over the Satisfiable Ensemble will finally allow us to characterize the solutions.

Entropic term

The entropic term of (5.24),

$$\mathcal{S}[c(\cdot)] \equiv - \sum_{\vec{\tau}} c(\vec{\tau}) \log c(\vec{\tau}) \quad (5.39)$$

can be computed by means of the following identity:

$$x \log x = \left. \frac{dx^{p+1}}{dp} \right|_{p=0}. \quad (5.40)$$

We obtain:

$$\mathcal{S}[c(\cdot)] = - \sum_{\vec{\tau}} \left. \frac{d}{dp} c(\vec{\tau})^{p+1} \right|_{p=0} = - \frac{d}{dp} \sum_{\vec{\tau}} c(\vec{\tau})^{p+1} \Big|_{p=0} \quad (5.41)$$

with

$$\sum_{\vec{\tau}} c(\vec{\tau})^{p+1} = \sum_{\vec{\tau}} \left\{ \int_{-\infty}^{\infty} dh R(h) \frac{\exp \left[\frac{\beta h}{2} \sum_{a=1}^n \tau^a \right]}{[2 \cosh \frac{\beta h}{2}]^n} \right\}^{p+1} \quad (5.42)$$

$$= \int_{-\infty}^{\infty} dh_1 \cdots dh_{p+1} R(h_1) \cdots R(h_{p+1}) \left[\frac{2 \cosh \left[\frac{\beta}{2} \sum_{j=1}^{p+1} h_j \right]}{2 \cosh \frac{\beta h_1}{2} \cdots \cosh \frac{\beta h_{p+1}}{2}} \right]^n. \quad (5.43)$$

We can now multiply by

$$1 = \int_{-\infty}^{\infty} d\hat{x} \delta \left(\hat{x} - \sum_{j=1}^{p+1} h_j \right) = \int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix(\hat{x} - \sum_{j=1}^{p+1} h_j)} \quad (5.44)$$

to obtain

$$\sum_{\vec{\tau}} c(\vec{\tau})^{p+1} = \int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x}} \left(2 \cosh \frac{\beta \hat{x}}{2} \right)^n \left[\int_{-\infty}^{\infty} dh \frac{R(h) e^{-ixh}}{(2 \cosh \frac{\beta h}{2})^n} \right]^{p+1}. \quad (5.45)$$

By taking the derivative as in (5.41) we find

$$\mathcal{S}[c(\cdot)] = - \int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x}} \left(2 \cosh \frac{\beta \hat{x}}{2} \right)^n \phi(x) \log \phi(x) \quad (5.46)$$

where

$$\phi(x) \equiv \int_{-\infty}^{\infty} dh \frac{R(h) e^{-ixh}}{(2 \cosh \frac{\beta h}{2})^n}. \quad (5.47)$$

In the limit $\beta \rightarrow \infty$, $n \rightarrow 0$ with finite $\nu = \beta n$ we have

$$\lim_{n \rightarrow 0} \left[2 \cosh \frac{\nu h}{2n} \right]^n = e^{\nu \frac{|h|}{2}} \quad (5.48)$$

and

$$\mathcal{S}[c(\cdot)] = - \int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x} + \nu \frac{|\hat{x}|}{2}} \phi(x) \log \phi(x) \quad (5.49)$$

with

$$\phi(x) = \int_{-\infty}^{\infty} dh e^{-ixh - \nu \frac{|h|}{2}} R(h). \quad (5.50)$$

Energetic term

For the second term in (5.24), we have

$$\mathcal{E}[c(\cdot)] \equiv \alpha \log \sum_{\vec{\tau}_1, \dots, \vec{\tau}_k} c(\vec{\tau}_1) \cdots c(\vec{\tau}_k) \mathcal{E}(\vec{\tau}_1, \dots, \vec{\tau}_k) \quad (5.51)$$

$$= \alpha \log \sum_{\vec{\tau}_1, \dots, \vec{\tau}_k} c(\vec{\tau}_1) \cdots c(\vec{\tau}_k) \exp \left\{ -\beta \sum_{a=1}^n \prod_{j=1}^k \delta(\tau_j^a, 1) \right\} \quad (5.52)$$

where I have simplified the expression (5.19) of the effective coupling \mathcal{E} taking profit from the sum over $\vec{\tau}_i$. Substitution of replica symmetric *ansatz* (5.29) gives

$$\begin{aligned} \mathcal{E}[c(\cdot)] &= \alpha \log \sum_{\vec{\tau}_1, \dots, \vec{\tau}_k} \int_{-\infty}^{\infty} dh_1 \cdots dh_k R(h_1) \cdots R(h_k) \times \\ &\times \frac{\exp \left[\beta \frac{h_1}{2} \sum_{a=1}^n \tau_1^a \right]}{\left(2 \cosh \frac{\beta h_1}{2} \right)^n} \cdots \frac{\exp \left[\beta \frac{h_k}{2} \sum_{a=1}^n \tau_1^a \right]}{\left(2 \cosh \frac{\beta h_k}{2} \right)^n} \exp \left\{ -\beta \sum_{a=1}^n \prod_{j=1}^k \delta(\tau_j^a, 1) \right\} \end{aligned} \quad (5.53)$$

$$\begin{aligned} &= \alpha \log \int_{-\infty}^{\infty} dh_1 \cdots dh_k \frac{R(h_1)}{\left(2 \cosh \frac{\beta h_1}{2} \right)^n} \cdots \frac{R(h_k)}{\left(2 \cosh \frac{\beta h_k}{2} \right)^n} \times \\ &\times \left\{ \sum_{\vec{\tau}} \exp \beta \left[\sum_{j=1}^k \frac{h_j}{2} \tau_j - \mathbb{I}[\tau_1 = \cdots = \tau_k = 1] \right] \right\}^n. \end{aligned} \quad (5.54)$$

As $\beta \rightarrow \infty$ the sum over τ is dominated by the term which maximizes the square parenthesis in (5.54), while the hyperbolic cosines are given by (5.48), so that:

$$\mathcal{E}[c(\cdot)] = \alpha \log \int_{-\infty}^{\infty} dh_1 \cdots dh_k R(h_1) \cdots R(h_k) e^{\nu \Phi(\mathbf{h})} \quad (5.55)$$

with $\mathbf{h} \equiv (h_1, \dots, h_k)$ and

$$\Phi(\mathbf{h}) = \max_{\tau \in \{-1, 1\}^k} \frac{1}{2} \sum_{j=1}^k (\tau_j h_j - |h_j|) - \mathbb{I}[\tau, \mathbf{1}] \quad (5.56)$$

$$= \begin{cases} -\min\{1, h_1, \dots, h_k\} & \text{if } h_j > 0 \ \forall j \\ 0 & \text{otherwise} \end{cases}. \quad (5.57)$$

The free energy functional we obtain, putting \mathcal{E} and \mathcal{S} together, is:

$$\mathcal{F}[R(\cdot), \nu, \alpha] = - \int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x} + \nu \frac{|\hat{x}|}{2}} \phi(x) \log \phi(x) + \alpha \log \int_{-\infty}^{\infty} dh_1 \cdots dh_k R(h_1) \cdots R(h_k) e^{\nu \Phi(\mathbf{h})} \quad (5.58)$$

with $\phi(x)$ and $\Phi(\mathbf{h})$ defined in (5.50) and (5.57).

5.2.4 Saddle point equations

We are now in position to derive the saddle point equations that will determine $R(h)$ from the extremality condition (5.27) for $\mathcal{F}[R(\cdot), \nu, \alpha]$, subject to the normalization condition (5.30), which we write as

$$\frac{\delta}{\delta R(\cdot)} \left\{ \mathcal{F}[R(\cdot), \nu, \alpha] + \lambda \left[\int R(h) dh - 1 \right] \right\} = 0 \quad (5.59)$$

$$\Leftrightarrow - \int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x} + \frac{1}{2}\nu|\hat{x}| - ixh - \frac{1}{2}\nu|h|} [1 + \log \phi(x)] + \frac{\alpha k}{\mathcal{D}[R(\cdot)]} \int_{-\infty}^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) e^{\nu\Phi(h, h_2, \dots, h_k)} + \lambda = 0 \quad (5.60)$$

where

$$\mathcal{D}[R(\cdot)] \equiv \int_{-\infty}^{\infty} dh_1 \cdots dh_k R(h_1) \cdots R(h_k) e^{\nu\Phi(\mathbf{h})} \quad (5.61)$$

and where it should be noted that the integral over the fields h_j in (5.60) starts with h_2 : both terms are functions of h , and they must be identically null.

In principle, the symmetry condition $R(h) = R(-h)$ should also be imposed, by means of a second Lagrange multiplier. However, it suffices to restrict the range over which (5.60) defines $R(h)$ to positive values of h and define $R(-h) \equiv R(h)$ with $h > 0$.

Because of the definition (5.57) of $\Phi(\mathbf{h})$, it is convenient to write the integral over the fields in (5.60) over \mathbb{R}^+ only. This can be done by noticing that if one of the h_i is negative then $\Phi(\mathbf{h}) = 0$, so that

$$\begin{aligned} & \int_{-\infty}^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) e^{\nu\Phi(h, h_2, \dots, h_k)} \\ &= \int_{-\infty}^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) + \\ & \quad + \int_0^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) \left(e^{\nu\Phi(h, h_2, \dots, h_k)} - 1 \right) \end{aligned} \quad (5.62)$$

$$= 1 - \frac{1}{2^{k-1}} + \int_0^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) e^{-\nu \min\{1, h, h_2, \dots, h_k\}} \quad (5.63)$$

because of the normalization and the symmetry of $R(h)$. We now multiply by the identity

$$\int_{-\infty}^{\infty} \frac{dy d\hat{y}}{2\pi} e^{iy[\hat{y} - \min\{1, h_2, \dots, h_k\}]} = 1 \quad (5.64)$$

to obtain

$$\begin{aligned} & \int_{-\infty}^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) e^{\nu\Phi(h, h_2, \dots, h_k)} \\ &= 1 - \frac{1}{2^{k-1}} + \int_{-\infty}^{\infty} \frac{dy d\hat{y}}{2\pi} e^{-\nu \min\{h, \hat{y}\} - iy\hat{y}} \times \\ & \quad \times \int_0^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) e^{iy \min\{1, h_2, \dots, h_k\}}. \end{aligned} \quad (5.65)$$

Notice that

$$\min\{h, \hat{y}\} = \frac{1}{2} [h + \hat{y} - |h - \hat{y}|] \quad (5.66)$$

so the exponent in the first integral of the previous equation can be written as

$$- iy\hat{y} - \frac{1}{2}\nu(h + \hat{y}) + \frac{1}{2}\nu|h - \hat{y}| \quad (5.67)$$

and changing the integration variables to $x = y - \frac{i}{2}\nu$ and $\hat{x} = h - \hat{y}$ we obtain

$$\begin{aligned} & \int_{-\infty}^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) e^{\nu \Phi(h, h_2, \dots, h_k)} \\ &= 1 - \frac{1}{2^{k-1}} + \int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x} + \frac{1}{2}\nu|\hat{x}| - ixh - \frac{1}{2}\nu h} \times \\ & \quad \times \int_0^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) e^{i(x + \frac{i}{2}\nu) \min\{1, h_2, \dots, h_k\}}. \end{aligned} \quad (5.68)$$

The exponent in the integral over $dx d\hat{x}$ is the same as in the first term of (5.60), and

$$\int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x} + \frac{1}{2}\nu|\hat{x}| - ixh - \frac{1}{2}\nu h} = \int_{-\infty}^{\infty} d\hat{x} \delta(\hat{x} - h) e^{\frac{1}{2}\nu|\hat{x}| - \frac{1}{2}\nu h} = 1 \quad (5.69)$$

since $h > 0$, so we can collect all the terms in (5.60) under the same integral. Let us define the following functions

$$K(h, x) = \int_{-\infty}^{\infty} \frac{d\hat{x}}{2\pi} e^{ix\hat{x} + \frac{1}{2}\nu|\hat{x}| - ixh - \frac{1}{2}\nu|h|}, \quad (5.70)$$

$$Q(x) = \int_0^{\infty} dh_2 \cdots dh_k R(h_2) \cdots R(h_k) e^{ix \min\{1, h_2, \dots, h_k\}} \quad (5.71)$$

in terms of which the saddle point equation (5.60) becomes

$$\int_{-\infty}^{\infty} dx K(h, x) \left\{ -[1 + \log \phi(x)] + \frac{\alpha k}{\mathcal{D}[R(\cdot)]} \left[1 - \frac{1}{2^{k-1}} + Q\left(x + \frac{i}{2}\nu\right) \right] + \lambda \right\} = 0. \quad (5.72)$$

A solution to this equation is obtained if the curly bracket vanishes identically. In that case, inverting (5.50) we obtain

$$R(h) = \int_{-\infty}^{\infty} \frac{dx}{2\pi} e^{ixh + \frac{1}{2}\nu h} \phi(x) \quad (5.73)$$

$$= \int_{-\infty}^{\infty} \frac{dx}{2\pi} \exp \left\{ ixh + \frac{1}{2}\nu h - 1 + \frac{\alpha k}{\mathcal{D}[R(\cdot)]} \left[1 - \frac{1}{2^{k-1}} + Q\left(x + \frac{i}{2}\nu\right) \right] + \lambda \right\}. \quad (5.74)$$

5.2.5 Distribution of fields

We are now in position to determine the distribution of fields $R(h)$ that satisfies the saddle point equation (5.74). Since this is a functional equation, its resolution is greatly simplified by making some assumption on the form of the function. I shall consider the following *ansatz* for $R(h)$,

$$R(h) = \sum_{p=-\infty}^{\infty} r_p \delta(h - p) \quad (5.75)$$

where only integer values of h are considered. I shall later prove that a more general form in which fractional values are considered reduces to this, suggesting that this is the only solution.

With this assumption (5.74) becomes an equation for the coefficients $\{r_p\}$. Let us begin from (5.61) by computing

$$\mathcal{D}[R(\cdot)] = \int_{-\infty}^{\infty} dh_1 \cdots dh_k R(h_1) \cdots R(h_k) e^{\nu \Phi(\mathbf{h})} \quad (5.76)$$

$$= \sum_{p_1, \dots, p_k} r_{p_1} \cdots r_{p_k} \times 1 + \sum_{p_1, \dots, p_k} r_{p_1} \cdots r_{p_k} (e^{-\nu} - 1) \quad (5.77)$$

$$= 1 + \left(\frac{1 - r_0}{2} \right)^k (e^{-\nu} - 1) \equiv D \quad (5.78)$$

where I used the fact that, for integer $\{h_j\}$, if some h_j is negative or null then $\Phi(\mathbf{h})$ is 0 and otherwise it is 1, and the symmetry and normalization of $R(h)$ which imply that $r_p = r_{-p}$ and $\sum_{p=-\infty}^{\infty} r_p = 1$.

Similarly for the term in Q in the exponent of (5.74), which can be written

$$Q\left(x + \frac{i}{2}\nu\right) = \int_0^\infty dh_2 \cdots dh_k R(h_2) \cdots R(h_k) \cos[x \min\{1, h_2, \dots, h_k\}] e^{-\frac{1}{2}\nu \min\{1, h_2, \dots, h_k\}} \quad (5.79)$$

$$= \sum_{p_2, \dots, p_k}^{0, \infty} r_{p_2} \cdots r_{p_k} \times 1 + \sum_{p_2, \dots, p_k}^{1, \infty} r_{p_2} \cdots r_{p_k} \left[\cos(x) e^{-\frac{1}{2}\nu} - 1 \right] \quad (5.80)$$

$$= \left(\frac{1+r_0}{2} \right)^{k-1} + \left(\frac{1-r_0}{2} \right)^{k-1} \left[\cos(x) e^{-\frac{1}{2}\nu} - 1 \right] \quad (5.81)$$

$$= A + B \cos(x) \quad (5.82)$$

with

$$A \equiv \left(\frac{1+r_0}{2} \right)^{k-1} - \left(\frac{1-r_0}{2} \right)^{k-1}, \quad (5.83)$$

$$B \equiv \left(\frac{1-r_0}{2} \right)^{k-1} e^{-\frac{1}{2}\nu}. \quad (5.84)$$

Substituting (5.78) and (5.82) into the saddle point equation (5.74) gives

$$R(h) = e^{\lambda' + \frac{1}{2}\nu|h|} \int_{-\infty}^{\infty} \frac{dx}{2\pi} \cos(xh) \exp \left\{ \alpha k \left[\frac{A}{D} + \frac{B}{D} \cos(x) \right] \right\} \quad (5.85)$$

where h can be positive or negative, and

$$\lambda' = \lambda - 1 + \frac{\alpha k}{D} \left[1 - \frac{1}{2^{k-1}} \right]. \quad (5.86)$$

This form is compatible with the *ansatz* (5.75), since it vanishes unless h is an integer, and we can invert it to obtain

$$r_p = e^{\lambda' + \frac{1}{2}\nu|h| + \alpha k \frac{A}{D}} \int_{-\pi}^{\pi} \frac{dx}{2\pi} e^{ixp} \exp \left[\alpha k \frac{B}{D} \cos(x) \right] \quad (5.87)$$

$$= e^{\lambda' + \frac{1}{2}\nu|h| + \alpha k \frac{A}{D}} I_p \left(\alpha k \frac{B}{D} \right) \quad (5.88)$$

where $I_p(x)$ is the modified Bessel function of integer order p . The value of λ' is determined by the normalization of $R(h)$, and we obtain

$$r_p = \frac{e^{\frac{1}{2}\nu|p|} I_p \left(\alpha k \frac{B}{D} \right)}{\sum_{q=-\infty}^{\infty} e^{\frac{1}{2}\nu|q|} I_q \left(\alpha k \frac{B}{D} \right)}. \quad (5.89)$$

In this formula, B depends on r_0 . It is therefore an equation for r_0 and, once solved for r_0 , and identity for all other values of p .

5.2.6 Ground state energy

Having obtained the explicit expression of the equilibrium distribution $R(h)$, we can compute the average value of the ground state energy density $\epsilon_0(\nu)$ for general ν .

Following (5.38), we write from the form of the free energy density functional (5.58)

$$\epsilon_0(\nu) = -\frac{\partial}{\partial \nu} \mathcal{F}[R(\cdot), \nu, \alpha] \quad (5.90)$$

$$\begin{aligned} &= \frac{1}{2} \int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x} + \frac{1}{2}\nu|\hat{x}|} \left\{ |\hat{x}| \phi(x) \log \phi(x) - [1 + \log \phi(x)] \int_{-\infty}^{\infty} dh e^{-ixh - \frac{1}{2}\nu|h|} |h| R(h) \right\} + \\ &\quad - \alpha \int_{-\infty}^{\infty} dh_1 \cdots dh_k \frac{R(h_1) \cdots R(h_k)}{\mathcal{D}[R(\cdot)]} \Phi(\mathbf{h}) e^{\nu \Phi(\mathbf{h})}. \end{aligned} \quad (5.91)$$

The integrals $dx d\hat{x}$ can be eliminated by means of the saddle point conditions (5.74) and (5.60), which give

$$\log \phi(x) = \lambda' + \alpha k \int_0^{\infty} dh_2 \cdots dh_k \frac{R(h_2) \cdots R(h_k)}{\mathcal{D}[R(\cdot)]} e^{(ix - \frac{1}{2}\nu) \min\{1, h_2, \dots, h_k\}} \quad (5.92)$$

$$\int_{-\infty}^{\infty} \frac{dx d\hat{x}}{2\pi} e^{ix\hat{x} + \frac{1}{2}\nu|\hat{x}|} [1 + \log \phi(x)] e^{-ixh - \frac{1}{2}\nu|h|} = \lambda \alpha k \int_0^{\infty} dh_2 \cdots dh_k \frac{R(h_2) \cdots R(h_k)}{\mathcal{D}[R(\cdot)]} e^{\nu \Phi(h, h_2, \dots, h_k)} \quad (5.93)$$

from which we obtain

$$\begin{aligned} \epsilon_0(\nu) &= - \int_0^{\infty} h R(h) dh + \frac{\alpha k}{4} \int_0^{\infty} dh_2 \cdots dh_k \frac{R(h_2) \cdots R(h_k)}{\mathcal{D}[R(\cdot)]} \min\{1, h_2, \dots, h_k\} + \\ &\quad + \alpha \int_0^{\infty} dh_1 \cdots dh_k \frac{R(h_1) \cdots R(h_k)}{\mathcal{D}[R(\cdot)]} \left[\frac{k}{2} \min\{1, h_2, \dots, h_k\} + (1-k) \min\{1, h_1, \dots, h_k\} \right] \times \\ &\quad \times e^{-\nu \min\{1, h_1, \dots, h_k\}}. \end{aligned} \quad (5.94)$$

This expression is valid independently on the form of $R(h)$. For the *ansatz* (5.75) we have

$$\begin{aligned} \epsilon_0(\nu) &= - \sum_{p=1}^{\infty} p r_p + \frac{\alpha k}{4} \sum_{p_2, \dots, p_k}^{1, \infty} \frac{r_{p_2} \cdots r_{p_k}}{D} + \frac{\alpha k}{4} r_0 \sum_{p_2, \dots, p_k}^{1, \infty} \frac{r_{p_2} \cdots r_{p_k}}{D} + \\ &\quad + \frac{\alpha k}{2} \sum_{p_1, \dots, p_k}^{1, \infty} \frac{r_{p_1} \cdots r_{p_k}}{D} e^{-\nu \min\{1, p_1\}} + \alpha(1-k) \sum_{p_1, \dots, p_k}^{1, \infty} \frac{r_{p_1} \cdots r_{p_k}}{D} e^{-\nu} \end{aligned} \quad (5.95)$$

$$= - \sum_{p=1}^{\infty} p r_p + \frac{\alpha k}{2D} \left(\frac{1-r_0}{2} \right)^{k-1} \left(\frac{1+r_0}{2} \right) + \frac{\alpha}{D} \left[\frac{k}{2} + (1-k) \right] \left(\frac{1-r_0}{2} \right)^k e^{-\nu} \quad (5.96)$$

$$= - \sum_{p=1}^{\infty} p r_p + \frac{\alpha k}{2} \frac{B}{D} \left(\frac{1+r_0}{2} \right) e^{\nu/2} + \alpha \left(1 - \frac{k}{2} \right) \frac{B}{D} \left(\frac{1-r_0}{2} \right) e^{-\nu/2} \quad (5.97)$$

where the term corresponding to $p_1 = 0$ in the first term of the second line of (5.94) has an extra factor 1/2 coming from the integral $\int_0^{\infty} \delta(x) dx$. The sum in the last expression can be computed as

$$\sum_{p=1}^{\infty} p r_p = -\frac{\partial}{\partial \nu} \log \mathcal{J} \left(\alpha k \frac{B}{D}, \nu \right) \quad (5.98)$$

where

$$\mathcal{J}(x, \nu) \equiv \sum_{p=-\infty}^{\infty} e^{-\frac{1}{2}\nu|p|} I_p(x) = 2e^{x \cosh(\nu/2)} - I_0(x) - 2 \sum_{p=1}^{\infty} e^{-\nu/2} I_p(x) \quad (5.99)$$

converges very fast for $\nu > 0$.

Large ν expansion

I am going to show that the condition $\epsilon_0(\nu) = 0$, which corresponds to the selection of satisfiable formulæ from the ensemble $\mathcal{P}_{\text{Sat}}[\mathcal{F}]$, is obtained for $\nu \rightarrow \infty$.

Let me denote $\varepsilon = e^{-\nu}$ and, to first order in ε

$$G \equiv \frac{\alpha k}{2} \frac{B}{D} e^{\nu/2} = \frac{\alpha k}{2} \frac{\left(\frac{1-r_0}{2}\right)^{k-1}}{1 - \left(\frac{1-r_0}{2}\right)^k (1-\varepsilon)} = \mathcal{G} \left[1 - \varepsilon \frac{2\mathcal{G}}{\alpha k} \frac{1-r_0}{2} \right], \quad (5.100)$$

$$\mathcal{G} \equiv \mathcal{G}(r_0) \equiv \frac{\alpha k}{2} \frac{\left(\frac{1-r_0}{2}\right)^{k-1}}{1 - \left(\frac{1-r_0}{2}\right)^k}. \quad (5.101)$$

The Bessel functions $I_p(x)$ can be expanded for small x and $p \geq 0$ as

$$I_p(x) = \frac{x^p}{2^p p!} \left[1 + \frac{x^2}{4(p+1)} + O(x^4) \right] \quad (5.102)$$

and $I_{-p}(x) = I_p(x)$. Since from the definition (5.84) of B we have that it is $O(\varepsilon^{1/2})$ while from (5.78) we have $D = O(1)$, we can expand

$$\mathcal{J} \left(\alpha k \frac{B}{D}, \nu \right) = \sum_{p=-\infty}^{\infty} e^{\frac{1}{2}\nu|p|} I_p \left(\alpha k \frac{B}{D} \right) \quad (5.103)$$

$$= \sum_{p=-\infty}^{\infty} e^{\frac{1}{2}\nu|p|} \frac{\left(\frac{\alpha k}{2} \frac{B}{D}\right)^{|p|}}{2^{|p|} |p|!} \left[1 + \frac{\left(\frac{\alpha k}{2} \frac{B}{D}\right)^2}{4(|p|+1)} + O(\varepsilon^2) \right] \quad (5.104)$$

$$= \sum_{p=-\infty}^{\infty} \frac{G^{|p|}}{|p|!} \left[1 + \varepsilon \frac{G^2}{|p|+1} + O(\varepsilon^2) \right] \quad (5.105)$$

$$= 2e^G - 1 + \varepsilon (2Ge^G - G^2 - 2G) + O(\varepsilon^2). \quad (5.106)$$

We can then write, in the equation (5.89) for r_0 , to the leading order in ε :

$$r_0 = \frac{I_0(2Ge^{\nu/2})}{2e^G - 1 + \varepsilon (2Ge^G - G^2 - 2G) + O(\varepsilon^2)} \quad (5.107)$$

$$= \frac{1 + \varepsilon G^2 + O(\varepsilon^2)}{2e^G - 1 + \varepsilon (2Ge^G - G^2 - 2G) + O(\varepsilon^2)} \quad (5.108)$$

$$= \frac{1}{2e^G - 1} \left\{ 1 + \varepsilon G^2 + \frac{\varepsilon}{2e^G - 1} \left[2e^{\mathcal{G}} \frac{2\mathcal{G}}{\alpha k} \frac{1-r_0}{2} - 2\mathcal{G}e^{\mathcal{G}} + \mathcal{G}^2 + 2\mathcal{G} \right] + O(\varepsilon^2) \right\} \quad (5.109)$$

$$\equiv F_0(r_0) + \varepsilon F_1(r_0) + O(\varepsilon^2). \quad (5.110)$$

Let me define

$$\rho_0 = \lim_{\nu \rightarrow \infty} r_0, \quad (5.111)$$

$$\rho_1 = \lim_{\nu \rightarrow \infty} \frac{r_0 - \rho_0}{\varepsilon} \quad (5.112)$$

so that $r_0 = \rho_0 + \varepsilon \rho_1 + o(\varepsilon)$. The value of ρ_0 is determined by the equation

$$\rho_0 = \frac{1}{2e^{\mathcal{G}(\rho_0)} - 1}. \quad (5.113)$$

The value of ρ_1 is obtained by developing (5.110) around ρ_0 :

$$\rho_0 + \varepsilon \rho_1 = \rho_0 + F'_0(\rho_0) \varepsilon \rho_0 + \varepsilon F_1(\rho_0) \quad (5.114)$$

which gives

$$\rho_1 = \frac{F_1(\rho_0)}{1 - F'_0(\rho_0)} \quad (5.115)$$

In order to write the average ground state energy for large ν we also need to compute

$$\sum_{p=1}^{\infty} p e^{\frac{1}{2}\nu p} I_p \left(\frac{\alpha k}{2} \frac{B}{D} \right) = \sum_{p=1}^{\infty} \frac{G^p}{(p-1)!} \left[1 + \frac{\varepsilon G^2}{p+1} + O(\varepsilon^2) \right] \quad (5.116)$$

$$= G e^G + \varepsilon G (1 - e^G + G e^G) + O(\varepsilon^2). \quad (5.117)$$

Using these expansions in the expression for the average ground state energy (5.97) we obtain, after some algebra,

$$\epsilon_0(\nu) = -\frac{G e^G + \varepsilon G (1 - e^G + G e^G)}{2e^G - 1 + \varepsilon (2G e^G - G^2 - 2G)} + G \frac{1+r_0}{2} + \varepsilon \left(\frac{2}{k} - 1 \right) G \frac{1-r_0}{2} \quad (5.118)$$

$$= -\mathcal{G} e^{\mathcal{G}} \rho_0 \left\{ 1 - \varepsilon \frac{2\mathcal{G}^2}{\alpha k} \frac{1-\rho_0}{2} \left[\frac{1}{\mathcal{G}} - \rho_0 \right] + \varepsilon [e^{-\mathcal{G}} - 1 + \mathcal{G}] - \varepsilon \rho_0 [2\mathcal{G} e^{\mathcal{G}} - \mathcal{G}^2 - 2\mathcal{G}] \right\} + \\ + \mathcal{G} \rho_0 e^{\mathcal{G}} \left[1 - \varepsilon \frac{2\mathcal{G}}{\alpha k} \frac{1-\rho_0}{2} \right] + \varepsilon \left(\frac{2}{k} - 1 \right) \mathcal{G} \frac{1-\rho_0}{2} \quad (5.119)$$

$$= \varepsilon \mathcal{G} e^{\mathcal{G}} \rho_0 \left\{ -\frac{\mathcal{G}^2 \rho_0 (1-\rho_0)}{\alpha k} - (e^{-\mathcal{G}} - 1 + \mathcal{G}) + \rho_0 (2\mathcal{G} e^{\mathcal{G}} - \mathcal{G}^2 - 2\mathcal{G}) + \frac{1}{\rho_0 e^{\mathcal{G}}} \left(\frac{2}{k} - 1 \right) \frac{1-\rho_0}{2} \right\} \quad (5.120)$$

where everything except ε is $O(1)$ as $\nu \rightarrow \infty$. Notice that the term in ρ_1 does not contribute to the first order result in the end.

The conclusion of this somewhat tedious calculation is that

$$\epsilon_0(\nu) \underset{\nu \rightarrow \infty}{\sim} e^{-\nu}. \quad (5.121)$$

Therefore, in order to obtain the equilibrium distribution of fields for formulæ extracted from the Satisfiable Ensemble $\mathcal{P}_{\text{Sat}}[\mathcal{F}]$, it is sufficient to take the limit $\nu \rightarrow \infty$ in (5.89), giving

$$\rho_0 = \frac{1}{2e^{\mathcal{G}(\rho_0)} - 1}, \quad (5.122)$$

$$\rho_p \equiv \lim_{\nu \rightarrow \infty} r_p = \frac{\mathcal{G}(\rho_0)^{|p|}}{|p|!} \frac{1}{2e^{\mathcal{G}(\rho_0)} - 1} \quad (p \neq 0) \quad (5.123)$$

where $\mathcal{G}(\rho_0)$ is defined by (5.101) as

$$\mathcal{G}(\rho_0) = \frac{\alpha k}{2} \frac{\left(\frac{1-\rho_0}{2} \right)^{k-1}}{1 - \left(\frac{1-\rho_0}{2} \right)^k}. \quad (5.124)$$

For any k and α it is easy to solve (5.122) to find ρ_0 , and then use it to compute all other ρ_p , thus completely defining the distribution of fields $R(h)$. In the following we shall see that this is sufficient to characterize some very interesting properties of the solutions. We shall also return on the two *ansatz* we made to derive these results: the replica symmetric form (5.29) of $c(\cdot)$, and the integer-only form of $R(h)$ in (5.75), in the Section 5.5 about the stability of the solution.

5.3 Cavity formalism for the fields distribution

The results of the previous Section can be obtained in a rather more straightforward way, at the price of some more assumptions.

Let us consider a formula over $N - 1$ variables, and let us add a new variable, which will appear in ℓ_+ new clauses as a non-negated literal, and in ℓ_- as a negated one. For random formulæ from the Uniform Ensemble, ℓ_+ and ℓ_- will be random variables with independent poissonian distribution

$$p_L(\ell) = \frac{(\alpha'k/2)^\ell}{\ell!} e^{-\alpha'k/2} \quad (5.125)$$

where α' is some constant that we shall determine later.

Let us denote by $1 - \rho_0$ the probability that an “old” variable is constrained, i.e. if it changes value some existing clause will be violated. Then, the new variable will be constrained if and only if all the $k - 1$ other variables in the clause are constrained, and if they appear with the “wrong” sign in the new clause. The probability for this to happen is

$$q = \left(\frac{1 - \rho_0}{2} \right)^{k-1}. \quad (5.126)$$

The number of clauses that contain the new variable x or its negation \bar{x} and which constrain them, which I shall denote m_+ and m_- respectively, will be independent random variables with distribution

$$p_M(m) = \sum_{\ell=m}^{\infty} p_L(\ell) \binom{\ell}{m} q^m (1-q)^{\ell-m} \quad (5.127)$$

$$= \frac{(\alpha'kq/2)^m}{m!} e^{-\alpha'kq/2}. \quad (5.128)$$

I shall also introduce a weighted distribution, in which ℓ is the weight, for later use:

$$p_M^w(m) = \sum_{\ell=m}^{\infty} \ell p_L(\ell) \binom{\ell}{m} q^m (1-q)^{\ell-m} \quad (5.129)$$

$$= p_M(m) \left[m + \frac{\alpha'k}{2} (1-q) \right] \quad (5.130)$$

(notice that this is *not* normalized, since $\sum_{m=0}^{\infty} p_M^w(m) = \alpha'k/2$).

The m_+ clauses that constrain x will be satisfied if $x = \text{TRUE}$, while the m_- clauses that constrain \bar{x} will be satisfied if $x = \text{FALSE}$. The minimal increase in energy after the addition of x to the formula is therefore

$$\Delta E = \min\{m_+, m_-\}. \quad (5.131)$$

Let me define the “magnetic field” h as the difference $m_+ - m_-$. Both ΔE and h will be random variables, with joint distribution

$$P(\Delta E, h) = \sum_{m_+=0}^{\infty} p_M(m_+) \sum_{m_-=0}^{\infty} p_M(m_-) \delta_{\Delta E, \min\{m_+, m_-\}} \delta_{h, m_+ - m_-}. \quad (5.132)$$

In the spirit of Paragraph 5.2.3, I am going to weight each possible new formula with a factor $e^{-\nu \Delta E}$. The probability that the new variable is subject to a field $h = p \in \mathbb{Z}$ is then

$$r_p(\nu) = \frac{\sum_{\Delta E \geq 0} P(\Delta E, p) e^{-\nu \Delta E}}{\sum_{\Delta E \geq 0} \sum_{m=-\infty}^{\infty} P(\Delta E, m) e^{-\nu \Delta E}}. \quad (5.133)$$

In order to restrict the computation to satisfiable formulæ, let us take the limit $\nu \rightarrow \infty$, so that only formulæ with $\Delta E = 0$ contribute. The probability that the new variable has zero field (i.e. that it is not constrained) is then

$$\rho_0 = \lim_{\nu \rightarrow \infty} r_0(\nu) = \frac{P(0, 0)}{\sum_{m=-\infty}^{\infty} P(0, m)} = \frac{1}{2e^{\alpha'kq/2} - 1} \quad (5.134)$$

and since q is a function of ρ_0 defined in (5.126), this expression is an equation which determines ρ_0 .

If we had no restrictions on the clauses added to the formula, their average number would be αk . However, we are restricting the ensemble to satisfiable formulæ only: some of the potential new clauses will have to be rejected, because they would make the formula UNSAT, and the average number of clauses effectively added will be

$$\langle l_+ + l_- \rangle = \frac{\sum_{m_+, m_-}^{0, \infty} [p_M^w(m_+) p_M(m_-) + p_M(m_+) p_M^w(m_-)] \delta_{0, \min\{m_+, m_-\}}}{\sum_{m=0}^{\infty} P(0, m)} \quad (5.135)$$

$$= \alpha' k \left[1 - \left(\frac{1 - \rho_0}{2} \right)^k \right]. \quad (5.136)$$

In order for α to be the clause to variable ratio of the formula, we must impose

$$\alpha = \alpha' \left[1 - \left(\frac{1 - \rho_0}{2} \right)^k \right] \quad (5.137)$$

which determines α' .

Multiplying on both sides by $kq/2$ and recalling the definition of q we obtain

$$\frac{\alpha' k q}{2} = \frac{\alpha k}{2} \frac{\left(\frac{1 - \rho_0}{2} \right)^{k-1}}{1 - \left(\frac{1 - \rho_0}{2} \right)^k} \quad (5.138)$$

which, compared with (5.124), gives

$$\mathcal{G}(\rho_0) = \frac{\alpha' k q}{2}. \quad (5.139)$$

The equation (5.134) for ρ_0 is then

$$\rho_0 = \frac{1}{2e^{\mathcal{G}(\rho_0)} - 1} \quad (5.140)$$

which is exactly the same as (5.113).

Notice that the distribution that we have computed is the distribution of the *cavity* fields, i.e. the fields acting on the new variable and generated by the old ones. *A priori* this distribution is different from that of the real fields, which include the effect of the new clauses on the values of the old variables (and therefore of the fields they induce). The distribution we are interested in is the one of the real fields, which is what we have computed by means of the replica calculation, not the distribution of cavity fields. However, it can be shown that these two distributions coincide in the case when they are poissonian. I shall now prove that this is indeed so, at least in the limit of large α .

The generating function $\mathbf{g}(x)$ of the distribution of variable occurrences $\ell_+ + \ell_-$ over satisfiable formulæ, i.e. such that $\min\{m_+, m_-\} = 0$, can be computed as

$$\begin{aligned} \mathbf{g}(x) &= \sum_{m_+=0}^{\infty} \sum_{\ell_+=m_+}^{\infty} p_L(\ell_+) \binom{\ell_+}{m_+} q^{m_+} (1-q)^{\ell_+-m_+} \times \\ &\quad \times \sum_{m_-=0}^{\infty} \sum_{\ell_-=m_-}^{\infty} p_L(\ell_-) \binom{\ell_-}{m_-} q^{m_-} (1-q)^{\ell_--m_-} \times x^{\ell_++\ell_-} \delta_{0, \min\{m_+, m_-\}} \end{aligned} \quad (5.141)$$

$$= e^{\alpha' k(x-1)(1-q)} \frac{2e^{\alpha' k x q/2} - 1}{2e^{\alpha' k q/2} - 1}. \quad (5.142)$$

For $\alpha \rightarrow \infty$ we see from (5.134) that $\rho_0 \rightarrow 0$ and from (5.137) that $\alpha' = O(\alpha)$, so that

$$\mathbf{g}(x) = e^{\alpha' k(x-1)(1-q/2)} + e^{-O(\alpha)} = e^{\alpha k(x-1)} \quad (5.143)$$

which is the generating function of a poissonian distribution of parameter αk .

The conclusion of this Section is that the interpretation of the field h as the number of clauses that are violated if a variable is flipped is correct, and the distribution of fields $R(h)$ is the distribution over the variables and the formulæ from \mathcal{P}_{Sat} of their values.

5.4 Comparison of \mathcal{P}_{Sat} and $\mathcal{P}_{\text{Plant}}$ at large α

I am now going to use the distribution of fields computed in Section (5.2) to show that, for $\alpha \rightarrow \infty$, the statistical properties of formulæ extracted from \mathcal{P}_{Sat} coincide with those of formulæ from $\mathcal{P}_{\text{Plant}}$.

For $\alpha \rightarrow \infty$ the solution to (5.122), (5.123) and (5.124) is

$$\rho_0 = \frac{1}{2e^\gamma - 1}, \quad (5.144)$$

$$\rho_p = \frac{\gamma^{|p|}}{|p|!} \frac{1}{2e^\gamma - 1}, \quad (5.145)$$

$$\gamma \equiv \mathcal{G}(0) = \frac{\alpha k}{2^k - 1}. \quad (5.146)$$

Since $\gamma = O(\alpha)$, this means that the fraction of variables that are not constrained is $\rho_0 = e^{-O(\alpha)}$: the solutions to a satisfiable formula at large α are all very similar to each other. Moreover, the average value of the fields is $O(\gamma) = O(\alpha)$, so the constrained variables have strong fields that force them to the correct assignment.

5.4.1 Distribution of fields

I shall now compute the distribution of fields for formulæ extracted from the Planted Ensemble $\mathcal{P}_{\text{Plant}}$.

Let us consider a configuration X , and a random clause \mathcal{C} satisfied by X . If one variable x_i is flipped, what is the probability q that \mathcal{C} is no longer satisfied? It is the product of the probability that \mathcal{C} contains x_i , which is k/N , times the probability that all the other literals in the clause have been chosen with the wrong sign, which is $1/(2^k - 1)$

$$q = \frac{k}{N} \frac{1}{2^k - 1}. \quad (5.147)$$

The number p of such clauses will be a random variable, with a binomial distribution $P(p)$ of parameter q

$$P(p) = \binom{M}{p} q^p (1 - q)^{M-p}. \quad (5.148)$$

For $N \rightarrow \infty$ this reduces to a poissonian of parameter $\alpha k/(2^k - 1)$, which is γ defined in (5.146),

$$P(p) \underset{N \rightarrow \infty}{\sim} e^{-\gamma} \frac{\gamma^p}{p!}. \quad (5.149)$$

In a random configuration X , half the variables will be TRUE, giving rise to positive fields, and half will be FALSE, giving negative fields. The distribution of fields, i.e. of the number of satisfied clauses that are violated if a variable is flipped, with the plus sign if that variable is TRUE and minus otherwise, is

$$\rho_p^{\text{Plant}} = \delta_{p,0} e^{-\gamma} + (1 - \delta_{p,0}) \frac{1}{2} e^{-\gamma} \frac{\gamma^{|p|}}{|p|!}. \quad (5.150)$$

Comparing with (5.145) we see that the two distributions of fields corresponding to the Satisfiable Ensemble at large α and to the Planted Ensemble differ by terms $e^{-O(\alpha)}$.

5.4.2 Correlation between field and number of occurrences

Not only the typical magnitude of the fields in formulæ from \mathcal{P}_{Sat} is of order α at large α , but it is correlated to a bias in the distribution of the relative number of occurrences of variables and their negations, as I shall prove with the following computation.

In order for a formula to be satisfiable, there must be no variable that receive contradictory messages, i.e. which is constrained by some clauses to be TRUE and by some other to be FALSE. If we assume that the field on the variable is $h > 0$, this means that the number m_- of clauses that constrain it to be FALSE must be 0, while the number m_+ of clauses that constrain it to be TRUE will be positive or null.

Let us denote by $\langle \ell_+ \rangle_{h>0}$ the average number of occurrences of such a variable in clauses where it is not negated, and by $\langle \ell_- \rangle_{h>0}$ the corresponding number for its negation. These will be random variables whose distribution can be expressed in terms of (5.128) and (5.130) as

$$\langle \ell \rangle_{h>0} = \frac{\sum_{m_+ \geq 1} p_M^w(m_+) p_M(0)}{\sum_{m_+ \geq 1} p_M(m_+) p_M(0)} \quad (5.151)$$

where in the numerator $p_M(0)$ is the probability that the number of clauses sending a negative message to the variable is 0, $p_M^w(m_+)$ is proportional to the average number of occurrences of the variable conditioned on the message it receives being positive, and the denominator is a normalization.

Using the explicit distributions (5.128) and (5.130) we have

$$\langle \ell_+ \rangle_{h>0} = \frac{\sum_{m_+ \geq 1} \frac{(\alpha' k q / 2)^{m_+}}{m_+!} e^{-\alpha' k q / 2} \left[m_+ + \frac{\alpha' k}{2} (1 - q) \right] \times e^{-\alpha' k q / 2}}{\sum_{m_+ \geq 1} \frac{(\alpha' k q / 2)^{m_+}}{m_+!} e^{-\alpha' k q / 2} \times e^{-\alpha' k q / 2}} \quad (5.152)$$

$$= \frac{\alpha' k}{2} \left[\frac{1 - (1 - q) e^{-\mathcal{G}}}{1 - e^{-\mathcal{G}}} \right] \quad (5.153)$$

$$= \frac{\alpha k}{2} \frac{1}{1 - 2^{-k}} + e^{-O(\alpha)}, \quad (5.154)$$

$$\langle \ell_- \rangle_{h>0} = \frac{\sum_{m_+ \geq 1} \frac{(\alpha' k q / 2)^{m_+}}{m_+!} e^{-\alpha' k q / 2} \times e^{-\alpha' k q / 2} \left[\frac{\alpha' k}{2} (1 - q) \right]}{\sum_{m_+ \geq 1} \frac{(\alpha' k q / 2)^{m_+}}{m_+!} e^{-\alpha' k q / 2} \times e^{-\alpha' k q / 2}} \quad (5.155)$$

$$= \frac{\alpha' k}{2} (1 - q) \quad (5.156)$$

$$= \frac{\alpha k}{2} \frac{1 - 2^{-(k-1)}}{1 - 2^{-k}} + e^{-O(\alpha)} \quad (5.157)$$

from which we obtain the average value of the *bias*

$$\frac{\langle \ell_+ \rangle_{h>0} - \langle \ell_- \rangle_{h>0}}{\langle \ell_+ \rangle_{h>0} + \langle \ell_- \rangle_{h>0}} = \frac{1}{2^k - 1} + e^{-O(\alpha)}. \quad (5.158)$$

Therefore variables with positive field appear more frequently non-negated than negated. Of course, the opposite is true for variables with negative field.

The same computation can be easily performed for formulæ from the Planted Ensemble. Given a configuration X and k indices of variables composing a clause, out of the 2^k possible choices of the negations of the corresponding literals only $2^k - 1$ will give satisfied clauses. If a variable x is TRUE in X , then the number of satisfied clauses in which it appears non-negated is 2^{k-1} , corresponding to the random choices of the signs of the other literals; the number of clauses in which it appears negated, however, will be smaller, as at least one of the other literals must have the proper sign to satisfy the clause, giving $2^{k-1} - 1$ possible choices.

Since the average numbers of occurrences of x and \bar{x} are proportional to these probabilities, we shall have

$$\frac{\langle \ell_+ \rangle_{\text{Plant}} - \langle \ell_- \rangle_{\text{Plant}}}{\langle \ell_+ \rangle_{\text{Plant}} + \langle \ell_- \rangle_{\text{Plant}}} = \frac{1/2^{k-1} - 1/(2^{k-1} - 1)}{1/2^{k-1} + 1/(2^{k-1} - 1)} \quad (5.159)$$

$$= \frac{1}{2^k - 1}. \quad (5.160)$$

Comparing with (5.158), we see that the distribution of the bias in the Planted Ensemble is the same as in the Satisfiable Ensemble at large α , up to terms $e^{-O(\alpha)}$.

5.4.3 Finite energy results

The results of the two previous paragraphs extend to formulæ with small positive energy, i.e. which are *not* satisfiable.

The average value of the ground state energy, given by (5.120), greatly simplifies for large α , giving

$$\epsilon_0(\nu) = \frac{\gamma}{k} [1 + O(\gamma^2 e^{-\gamma})] e^{-\nu} \quad (5.161)$$

with $\gamma = O(\alpha)$ defined in (5.146).

The computation of the bias (5.151) can be generalized to finite large values of ν by including positive values of m_- , weighted with a factor $e^{-\nu m_-}$. To first order in $e^{-\nu}$ only $m_- = 1$ contributes and we have

$$\langle \ell \rangle_{h>0} = \frac{\sum_{m_+ \geq 1} p_M^w(m_+) \sum_{0 \leq m_- < m_+} p_M(m_-) e^{-\nu m_-}}{\sum_{m_+ \geq 1} p_M(m_+) \sum_{0 \leq m_- < m_+} p_M(m_-) e^{-\nu m_-}} \quad (5.162)$$

$$= \frac{\sum_{m_+ \geq 1} p_M^w(m_+) p_M(0) + \sum_{m_+ \geq 2} p_M^w(m_+) p_M(1) e^{-\nu}}{\sum_{m_+ \geq 1} p_M(m_+) p_M(0) + \sum_{m_+ \geq 2} p_M^w(m_+) p_M(1) e^{-\nu}} + O(e^{-2\nu}). \quad (5.163)$$

Computing the sums as for (5.151), we obtain

$$\frac{\langle \ell_+ \rangle_{h>0} - \langle \ell_- \rangle_{h>0}}{\langle \ell_+ \rangle_{h>0} + \langle \ell_- \rangle_{h>0}} = \frac{1}{2^k - 1} - \frac{\alpha k}{2(2^k - 1)^2} e^{-\nu} + O(\alpha^{-1}) + O(e^{-2\nu}), \quad (5.164)$$

where we can use (5.161) to eliminate $e^{-\nu}$ and obtain

$$\frac{\langle \ell_+ \rangle_{h>0} - \langle \ell_- \rangle_{h>0}}{\langle \ell_+ \rangle_{h>0} + \langle \ell_- \rangle_{h>0}} = \frac{1}{2^k - 1} \left[1 - \epsilon_0 k 2^k \left(\frac{1}{2} - \frac{1}{2^{k+1} - 2} - \frac{2^k}{\alpha k} \right) \right] + o(\epsilon_0). \quad (5.165)$$

We see that as long as $\epsilon_0 \ll 2^{-k}/k$ the bias remains of the same order as for satisfiable formulæ.

5.4.4 Algorithmic implications

In this section, I have shown that the distribution of fields ρ_p and the average bias obtained for formulæ extracted from the Planted Ensemble coincides with those for formulæ extracted from the Satisfiable Ensemble for large enough α , and that this extends to finite energy formulæ from the Uniform Ensemble, provided the energy is $\epsilon_0 \ll 2^{-k}/k$.

The demonstration of [75] of the convergence of WP is based on the following facts, which are proved for the Planted Ensemble:

- At large α , typical formulæ have a large *core*, i.e. a set of variables that take the same value in all the solutions to the formula. The fraction of variables that are not in the core is $e^{-O(\alpha)}$.

- The cavity fields corresponding to core variables and computed for satisfying assignments are of order $O(\alpha)$.
- Even for random assignments, the cavity fields are of order $O(\alpha)$. This is due to the fact that the value of core variables in satisfying assignments is correlated to a bias in the relative number of occurrences of the variable and its negation in the formula.

As we have seen in this Section, each of these properties holds as well for formulæ drawn from the Satisfiable Ensemble \mathcal{P}_{Sat} , provided α is large enough. This supports the conclusion that the convergence of WP should extend to \mathcal{P}_{Sat} . I therefore claim that Hypothesis 1_p , formulated at the end of Paragraph 5.1.4, is refuted by WP for any $p > 0$.

Moreover, a probabilistic version of Hypothesis 2 states that

Hypothesis 2_p For every fixed $\epsilon \geq 0$, even when α is an arbitrarily large constant (independent on N), there is no polynomial time algorithm that on most Random-3-SAT formulæ outputs TYPICAL and outputs NOT TYPICAL with probability p on formulæ with $(1 - \epsilon)M$ satisfiable clauses.

The finite energy results of Paragraph 5.4.3 support the claim that Hypothesis 2_p is refuted by WP for any $p > 0$ provided $\epsilon \ll 2^{-k}/k$.

5.5 Stability of the RS free energy

The conclusions of the previous sections are based on two *ansatz*: that the order parameter $c(\cdot)$ has the replica symmetric form (5.29), and that the distribution of fields $R(h)$ is non zero only for integer values of the fields, in (5.75).

In this Section, I shall support the claim that these two *ansatz* are correct. In order to do this, I shall prove that more general solutions for the saddle point equations that determine $R(h)$, which are non zero for fractional values of h , reduce to the *ansatz*, i.e. that the non-integer contributions vanish. Then I shall prove that the eigenvalues of the stability matrix of the saddle point equations computed for the replica symmetric form of $c(\cdot)$ are all negative for large enough α and $\nu \rightarrow \infty$. This does not prove that the *ansatz* corresponds to a global minimum, but only to a *local* one. In order to rule out the existence of other solutions to the saddle point equations, I shall prove that two real replicas of the formula necessarily have the same distribution of fields, and therefore must be in the same thermodynamic state, which is therefore unique.

5.5.1 Solutions with non-integer fields

Instead of the integer valued *ansatz* of (5.75), let us assume that $R(h)$ takes the more general form

$$R(h) = \sum_{p=-\infty}^{\infty} r_p \delta\left(h - \frac{p}{q}\right) \quad (5.166)$$

where q is an integer larger than 1. Substituting this assumption in the saddle point equations (5.74) gives the following *functional* equation

$$\sum_{p=-\infty}^{\infty} r_p \cos\left(x \frac{p}{q}\right) e^{-\frac{\nu|p|}{2q}} = \exp\left[\mu + \alpha k \sum_{j=1}^q A_j \cos\left(x \frac{p}{q}\right) e^{-\frac{\nu j}{2q}}\right] \quad (5.167)$$

which must be true for any x , where μ is a constant, and where

$$A_1 \equiv \frac{w^{k-1} - (w - r_1)^{k-1}}{1 - w^k}, \quad (5.168)$$

$$A_j \equiv \frac{(w - r_{j-1})^{k-1} - (w - r_j)^{k-1}}{1 - w^k} \quad (1 < j < q), \quad (5.169)$$

$$A_q \equiv \frac{(w - r_{p-1})^{k-1}}{1 - w^k}, \quad (5.170)$$

$$w \equiv \frac{1 - r_0}{2}. \quad (5.171)$$

The value of μ can be determined by taking $x = i\nu/2$ and then sending $\nu \rightarrow \infty$, which gives

$$\sum_{p=-\infty}^{\infty} r_p \frac{1 + \delta_{p,0}}{2} = \exp \left(\mu + \frac{\alpha k}{2} \sum_{j=1}^q A_j \right). \quad (5.172)$$

By taking instead $x = 0$ and sending $\nu \rightarrow \infty$ one also obtains that

$$r_0 = e^\mu. \quad (5.173)$$

Combining these two identities, we obtain an equation for r_0 :

$$r_0 = \frac{1}{2 \exp \left[\frac{\alpha k}{2} \frac{w^{k-1}}{1 - w^k} \right] - 1}. \quad (5.174)$$

Notice that this is exactly the same equation (5.122) and (5.124) that we have obtained with the *ansatz* of integer fields (5.75).

For $j = 1$ we have from (5.167):

$$r_1 = r_0 \frac{A_1}{2} \quad (5.175)$$

which can be written as

$$r_1 = \frac{r_0}{2} \frac{w^{k-1} - (w - r_1)^{k-1}}{1 - w^k}. \quad (5.176)$$

Notice that $r_1 = 0$ is a solution of this equation. The derivative with respect to r_1 of the right hand side is

$$\frac{r_0}{2} \frac{(k-1)(w - r_1)^{k-2}}{1 - w^k}. \quad (5.177)$$

When α is large, $r_0 = e^{-O(\alpha)}$ and $w = 1/2 - e^{-O(\alpha)}$. The possible range of value of r_1 goes from 0 to w (which is the probability of the field being positive, and therefore must be larger than or equal to r_1). For large enough α this derivative is much smaller than 1 for any of the possible values of r_1 , and therefore there cannot be another solution to (5.176).

A similar argument can be constructed for any of the coefficients r_p corresponding to fractional values of the field, showing that only integer values are admissible among rationals. Of course, this doesn't prove that other distributions $R(h)$ satisfying the saddle point equations and involving irrational fields cannot exist, but it is a rather strong indication that the *ansatz* (5.75) is correct.

5.5.2 Eigenvalues of the stability matrix

The stability matrix of the free energy (5.24) is defined as its second derivative,

$$\mathbf{M}_{\vec{\sigma}\vec{\tau}} = \frac{\partial^2 \mathcal{F}}{\partial c(\vec{\sigma}) \partial c(\vec{\tau})} \quad (5.178)$$

which gives:

$$\begin{aligned} \mathbf{M}_{\vec{\sigma}\vec{\tau}} &= -\frac{1}{c(\vec{\sigma})}\delta_{\vec{\sigma},\vec{\tau}} + \frac{\alpha k(k-1) \sum_{\vec{\sigma}_3 \dots \vec{\sigma}_k} c(\vec{\sigma}_3) \dots c(\vec{\sigma}_k) \mathcal{E}(\vec{\sigma}, \vec{\tau}, \vec{\sigma}_3, \dots, \vec{\sigma}_k)}{\sum_{\vec{\sigma}_1 \dots \vec{\sigma}_k} c(\vec{\sigma}_1) \dots c(\vec{\sigma}_k) \mathcal{E}(\vec{\sigma}_1, \dots, \vec{\sigma}_k)} + \\ &\quad - \frac{\alpha k^2 \sum_{\vec{\sigma}_2 \dots \vec{\sigma}_k} c(\vec{\sigma}_2) \dots c(\vec{\sigma}_k) \mathcal{E}(\vec{\sigma}, \vec{\sigma}_2, \dots, \vec{\sigma}_k) \sum_{\vec{\sigma}'_2 \dots \vec{\sigma}'_k} c(\vec{\sigma}'_2) \dots c(\vec{\sigma}'_k) \mathcal{E}(\vec{\tau}, \vec{\sigma}'_2, \dots, \vec{\sigma}'_k)}{[\sum_{\vec{\sigma}_1 \dots \vec{\sigma}_k} c(\vec{\sigma}_1) \dots c(\vec{\sigma}_k) \mathcal{E}(\vec{\sigma}_1, \dots, \vec{\sigma}_k)]^2}. \end{aligned} \quad (5.179)$$

The solution $c(\cdot)$ of the saddle point equations, given by the equations (5.29), (5.75) and (5.123), can be written as

$$c(\vec{\sigma}) = \frac{1}{2e^{\mathcal{G}} - 1} \left\{ \exp \left[\mathcal{G} e^{-\frac{\nu(1-s)}{2}} \right] + \exp \left[\mathcal{G} e^{-\frac{\nu(1+s)}{2}} \right] - 1 \right\} \quad (5.180)$$

where

$$s \equiv \frac{1}{n} \sum_{a=1}^n \sigma^a \quad (5.181)$$

and \mathcal{G} is defined in (5.124). In the limit $\nu \rightarrow \infty$ this reduces to

$$c(\vec{\sigma}) = \frac{1}{2e^{\mathcal{G}} - 1} [e^{\mathcal{G}} \delta_{|s|,1} + (1 - \delta_{|s|,1})] . \quad (5.182)$$

For large α this further simplifies, as $\mathcal{G} = O(\alpha)$ so that

$$c(\vec{\sigma}) = \frac{1}{2} \delta_{|s|,1} + e^{-O(\alpha)} . \quad (5.183)$$

We can now compute the sums that appear in the expression of \mathbf{M} . In order to do this, let me recall the definition of the effective interaction \mathcal{E} from (5.19):

$$\mathcal{A}_k \equiv \sum_{\vec{\sigma}_1 \dots \vec{\sigma}_k} c(\vec{\sigma}_1) \dots c(\vec{\sigma}_k) \mathcal{E}(\vec{\sigma}_1, \dots, \vec{\sigma}_k) \quad (5.184)$$

$$= \sum_{\vec{\sigma}_1, \dots, \vec{\sigma}_k} c(\vec{\sigma}_1) \dots c(\vec{\sigma}_k) \frac{1}{2^k} \sum_{q_1, \dots, q_k}^{\{-1,1\}} \exp \left\{ -\beta \sum_{a=1}^n \prod_{j=1}^k \delta(\sigma_j^a, q_j) \right\} . \quad (5.185)$$

In the limit $\beta \rightarrow \infty$, only the terms where the exponent vanish contribute. The value of \mathcal{E} is then 2^{-k} times the number of k -component vectors \underline{v} such that for any $j = 1, \dots, k$ and any $a = 1, \dots, n$ we have $v_j \neq \sigma_j^a$. Since the only $\vec{\sigma}$ that have a non-vanishing $c(\vec{\sigma})$ are $\{\sigma_a = 1 \ (\forall a = 1, \dots, n)\}$ and $\{\sigma_a = -1 \ (\forall a = 1, \dots, n)\}$, these n conditions are actually identical, and only one (out of the possible 2^k) vector \underline{v} is excluded.

The sum over the k vectors $\vec{\sigma}_j$ therefore has 2^k terms (corresponding to the 2 possible values of $\vec{\sigma}_j$), each of which has a factor 2^{-k} from the product of the $c(\cdot)$'s, and a factor $2^{-k} \times (2^k - 1)$ from the \mathcal{E} , so that

$$\mathcal{A}_k = 2^k \times \frac{1}{2^k} \times \frac{1}{2^k} (2^k - 1) = 1 - \frac{1}{2^k} . \quad (5.186)$$

In a very similar way,

$$\mathcal{A}_{k-1}(\vec{\sigma}) \equiv \sum_{\vec{\sigma}_2 \dots \vec{\sigma}_k} c(\vec{\sigma}_2) \dots c(\vec{\sigma}_k) \mathcal{E}(\vec{\sigma}, \vec{\sigma}_2, \dots, \vec{\sigma}_k) \quad (5.187)$$

$$= 2^{k-1} \times \frac{1}{2^{k-1}} \times \frac{1}{2^k} [2^k - (2 - \delta_{|s|,1})] \quad (5.188)$$

$$= 1 - \frac{1}{2^{k-1}} + \frac{\delta_{|s|,1}}{2^k} , \quad (5.189)$$

since if $|s| = 1$ all the columns in the matrix σ will be equal (and only one vector \underline{v} will be excluded), while if $|s| < 1$ there will be two column values (and correspondingly 2 vectors \underline{v} excluded).

Finally,

$$\mathcal{A}_{k-2}(\vec{\sigma}, \vec{\tau}) \equiv \sum_{\vec{\sigma}_3 \cdots \vec{\sigma}_k} c(\vec{\sigma}_3) \cdots c(\vec{\sigma}_k) \mathcal{E}(\vec{\sigma}, \vec{\tau}, \vec{\sigma}_3, \dots, \vec{\sigma}_k) \quad (5.190)$$

$$= 2^{k-2} \times \frac{1}{2^{k-2}} \times \frac{1}{2^k} [2^k - A(\vec{\sigma}, \vec{\tau})] \quad (5.191)$$

$$= 1 - \frac{A(\vec{\sigma}, \vec{\tau})}{2^k} \quad (5.192)$$

where $A(\vec{\sigma}, \vec{\tau})$ counts the number of different pairs, among the possible four which are $(1, 1)$, $(1, -1)$, $(-1, 1)$, $(-1, -1)$, that actually occur in the set $\{(\sigma^a, \tau^a) | a = 1, \dots, n\}$.

We can then substitute these expression in (5.179) to obtain, up to terms of order $e^{-O(\alpha)}$,

$$\mathbf{M}_{\vec{\sigma}\vec{\tau}} = -\frac{2e^{\mathcal{G}} \delta_{\vec{\sigma}, \vec{\tau}}}{e^{\mathcal{G}} \delta_{|s|, 1} + (1 - \delta_{|s|, 1})} + \frac{\alpha k(k-1) [2^k - A(\vec{\sigma}, \vec{\tau})]}{2^k - 1} + \quad (5.193)$$

$$- \frac{\alpha k^2 [2^k - 2 + \delta_{|s|, 1}] [2^k - 2 + \delta_{|t|, 1}]}{(2^k - 1)^2} \quad (5.194)$$

where t is defined for $\vec{\tau}$ as s for $\vec{\sigma}$. This matrix is invariant under the exchange of replica indices, and therefore it can be block-diagonalized in subspaces of well-defined replica symmetry.

In order to take into account the normalization constraint

$$\sum_{\vec{\sigma}} c(\vec{\sigma}) = 1, \quad (5.195)$$

it is convenient to decompose the dependency of $\mathcal{F}[c(\cdot)]$ in two, writing

$$\mathcal{F}[c(\vec{\sigma})] \equiv \mathcal{F}' \left[1 - \sum_{\vec{\sigma}} c'(\vec{\sigma}), c'(\vec{\sigma}) \right] \quad (5.196)$$

with $c'(\vec{\sigma}) = c(\vec{\sigma})$ for every $\vec{\sigma}$ except $\vec{\sigma} = \vec{1} \equiv (1, \dots, 1)$, and 0 otherwise, and where \mathcal{F}' is the functional defined by the previous identity. The stability matrix of \mathcal{F}' is then

$$\mathbf{M}'_{\vec{\sigma}\vec{\tau}} = \mathbf{M}_{\vec{\sigma}\vec{\tau}} - \mathbf{M}'_{\vec{\sigma}\vec{1}} - \mathbf{M}'_{\vec{1}\vec{\tau}} + \mathbf{M}'_{\vec{1}\vec{1}} \quad (5.197)$$

$$= -\frac{2e^{\mathcal{G}} \delta_{\vec{\sigma}, \vec{\tau}}}{e^{\mathcal{G}} \delta_{|s|, 1} + (1 - \delta_{|s|, 1})} - 2 - \frac{\alpha k(k-1)}{2^k - 1} [1 + A(\vec{\sigma}, \vec{\tau}) - A(\vec{1}, \vec{\tau}) - A(\vec{\sigma}, \vec{1})] + \quad (5.198)$$

$$- \frac{\alpha k^2 [\delta_{|s|, 1} - 2] [\delta_{|t|, 1} - 2]}{(2^k - 1)^2}.$$

In non-symmetric subspaces, $|s| \neq 1 \neq |t|$, and the previous equation becomes

$$\mathbf{M}'_{\vec{\sigma}\vec{\tau}} = -2e^{\mathcal{G}} \delta_{\vec{\sigma}, \vec{\tau}} - 2 - \frac{\alpha k(k-1)}{2^k - 1} [1 + A(\vec{\sigma}, \vec{\tau}) - A(\vec{1}, \vec{\tau}) - A(\vec{\sigma}, \vec{1})] - \frac{4\alpha k^2}{(2^k - 1)^2}. \quad (5.199)$$

The diagonal terms of this matrix are of order $O(e^\alpha)$, while the off-diagonal terms are of order $O(\alpha)$. The contribution of the off-diagonal terms to the eigenvalues will be given by 2^n terms, each of $O(\alpha)$. Since the contribution of the diagonal terms to the eigenvalues is of order $O(e^\alpha)$ and it is negative, this ensures that for large enough α all the eigenvalues will be negative. In this subspaces, the replica symmetric solution is therefore a local maximum.

In the symmetric subspace, all the diagonal elements of $\mathbf{M}'_{\vec{\sigma}\vec{\tau}}$ are of order $O(e^\alpha)$, *except* the term corresponding to $\vec{\sigma} = \vec{\tau} = -\vec{1}$: for this term, the exponential contributions vanish. However, we can then write

$$\mathbf{M}'_{\vec{\sigma}\vec{\tau}} = -2e^\alpha \delta_{\vec{\sigma}\vec{\tau}} + \mathbf{V}_{\vec{\sigma}\vec{\tau}} \quad (5.200)$$

and treat \mathbf{V} as a perturbation. For $\vec{\sigma} = \vec{\tau} = -\vec{1}$ the matrix element of \mathbf{V} is

$$\mathbf{V}_{-\vec{1}, -\vec{1}} = \mathbf{M}'_{-\vec{1}, -\vec{1}} = \mathbf{M}_{-\vec{1}, -\vec{1}} - \mathbf{M}_{-\vec{1}, \vec{1}} - \mathbf{M}_{\vec{1}, -\vec{1}} + \mathbf{M}_{\vec{1}, \vec{1}} \quad (5.201)$$

and from (5.179) this is equal to -4 , so it is negative.

The conclusion of this analysis is that, for α large enough, all the eigenvalues of the stability matrix of \mathcal{F} , computed for the $c(\cdot)$ which satisfies the replica symmetric saddle point equations, are negative, and therefore that this solution is *locally* stable.

5.5.3 Uniqueness of the solution

The conclusion from the previous Paragraph cannot rule out the existence of other solutions to the saddle point equations, which could possibly be the true *global* maximum of \mathcal{F} . I shall now provide an argument supporting that the saddle point equations have a unique solution, which is therefore the one found in Section 5.2.

Let us consider two real replicas of the system, i.e. two identical satisfiable formulæ. I shall indicate by α the thermodynamic state of the first replica, and by β that of the second one (the context will make it obvious when α refers to the clause to variable ratio of the formula). I want to study, with the cavity method, the joint probability for a variable of having a positive, negative or null field in the two states α and β , which I shall denote by the following quantities:

$$\begin{array}{ccc} p_{++}^{\alpha\beta} & p_{+0}^{\alpha\beta} & p_{+-}^{\alpha\beta} \\ p_{0+}^{\alpha\beta} & p_{00}^{\alpha\beta} & p_{0-}^{\alpha\beta} \\ p_{-+}^{\alpha\beta} & p_{-0}^{\alpha\beta} & p_{--}^{\alpha\beta} \end{array} \quad (5.202)$$

What I want to prove is that for large α :

- The off-diagonal terms become negligible, so that the fields are equal in the two states for most variables;
- The term $p_{00}^{\alpha\beta}$ is much smaller than $p_{++}^{\alpha\beta}$ and $p_{--}^{\alpha\beta}$.

The consequence of these two properties will be that most variables will be constrained to take the same values in the two states α and β , which is therefore a single, unique, thermodynamic state.

Distribution of the number of messages

Let us assume that a new variable is added to the formula, appearing non-negated in l_+ clauses and negated in l_- clauses. These will be two independent random variables with identical poissonian distribution of parameter $\alpha'k/2$, where α' is some constant which will be determined later.

A clause will send a message to a variable (that is, it will constrain it) if all the other variables in the clause are constrained (that is, have a non-zero field) and appear with the “wrong” sign in the

clause, which happens with probabilities

$$q^\alpha = \left[\frac{1 - (p_{0+}^{\alpha\beta} + p_{00}^{\alpha\beta} + p_{0-}^{\alpha\beta})}{2} \right]^{k-1}, \quad (5.203)$$

$$q^\beta = \left[\frac{1 - (p_{+0}^{\alpha\beta} + p_{00}^{\alpha\beta} + p_{-0}^{\alpha\beta})}{2} \right]^{k-1} \quad (5.204)$$

respectively in the states α and β .

For a given l_+ , the probability that in the state α the number clauses sending a message to the new variable is m_+^α is equal to

$$p_M^\alpha(m_+^\alpha | l_+) = \binom{l_+}{m_+^\alpha} (q^\alpha)^{m_+^\alpha} (1 - q^\alpha)^{l_+ - m_+^\alpha} \quad (5.205)$$

and identical distributions are valid for m_-^α for fixed l_- , and for the corresponding quantities in the state β .

The number of occurrences l_+ must be the same in the two states (the replicas are identical), and must be larger than m_+^α and m_+^β . The joint distribution of m_+^α and m_+^β is obtained by summing over the allowed values of l_+ :

$$\begin{aligned} p_M^{\alpha\beta}(m_+^\alpha, m_+^\beta) &= \sum_{l_+ = \max(m_+^\alpha, m_+^\beta)}^{\infty} \frac{1}{(l_+)!} \left(\frac{\alpha' k}{2} \right)^{l_+} e^{-\alpha' k/2} \times \binom{l_+}{m_+^\alpha} (q^\alpha)^{m_+^\alpha} (1 - q^\alpha)^{l_+ - m_+^\alpha} \\ &\quad \times \binom{l_+}{m_+^\beta} (q^\beta)^{m_+^\beta} (1 - q^\beta)^{l_+ - m_+^\beta} \end{aligned} \quad (5.206)$$

and similarly for the negative messages.

The joint probability of all messages is given by the product of the distributions of positive and negative messages, since they are independent:

$$\mathbb{P}[m_+^\alpha, m_+^\beta, m_-^\alpha, m_-^\beta] = p_M^{\alpha\beta}(m_+^\alpha, m_+^\beta) \times p_M^{\alpha\beta}(m_-^\alpha, m_-^\beta). \quad (5.207)$$

The values of $\{p_{++}^{\alpha\beta}, \dots, p_{--}^{\alpha\beta}\}$ are obtained from this distribution by summing over the appropriate ranges the values of m_\pm .

Selection of satisfiable formulæ

In order to have a satisfiable formula, no variable must receive contradictory messages. This means that the ranges to be considered in the sums to compute $\{p_{++}^{\alpha\beta}, \dots, p_{--}^{\alpha\beta}\}$ must be the following:

$$p_{00}^{\alpha\beta} : p_M^{\alpha\beta}(0, 0) \times p_M^{\alpha\beta}(0, 0) \quad (5.208)$$

$$p_{++}^{\alpha\beta} : p_M^{\alpha\beta}(m_+^\alpha, m_+^\beta) \times p_M^{\alpha\beta}(0, 0) \quad (5.209)$$

$$p_{--}^{\alpha\beta} : p_M^{\alpha\beta}(0, 0) \times p_M^{\alpha\beta}(m_-^\alpha, m_-^\beta) \quad (5.210)$$

$$p_{+-}^{\alpha\beta} : p_M^{\alpha\beta}(m_+^\alpha, 0) \times p_M^{\alpha\beta}(0, m_-^\beta) \quad (5.211)$$

$$p_{-+}^{\alpha\beta} : p_M^{\alpha\beta}(0, m_+^\beta) \times p_M^{\alpha\beta}(m_-^\alpha, 0) \quad (5.212)$$

$$p_{0+}^{\alpha\beta} : p_M^{\alpha\beta}(0, m_+^\beta) \times p_M^{\alpha\beta}(0, 0) \quad (5.213)$$

$$p_{0-}^{\alpha\beta} : p_M^{\alpha\beta}(0, 0) \times p_M^{\alpha\beta}(0, m_-^\beta) \quad (5.214)$$

$$p_{+0}^{\alpha\beta} : p_M^{\alpha\beta}(m_+^\alpha, 0) \times p_M^{\alpha\beta}(0, 0) \quad (5.215)$$

$$p_{-0}^{\alpha\beta} : p_M^{\alpha\beta}(0, 0) \times p_M^{\alpha\beta}(m_-^\alpha, 0) \quad (5.216)$$

where all the m_{\pm} are positive, and must be summed between 1 and infinity.

I therefore define:

$$S_0 \equiv p_M^{\alpha\beta}(0,0)^2, \quad (5.217)$$

$$S_1 \equiv \sum_{m_+^{\alpha}, m_+^{\beta}=1}^{\infty} p_M^{\alpha\beta}(m_+^{\alpha}, m_+^{\beta}) p_M^{\alpha\beta}(0,0) = \sum_{m_-^{\alpha}, m_-^{\beta}=1}^{\infty} p_M^{\alpha\beta}(0,0) p_M^{\alpha\beta}(m_-^{\alpha}, m_-^{\beta}), \quad (5.218)$$

$$S_2 \equiv \sum_{m_+^{\alpha}=1}^{\infty} \sum_{m_-^{\beta}=1}^{\infty} p_M^{\alpha\beta}(m_+^{\alpha}, 0) p_M^{\alpha\beta}(0, m_-^{\beta}) = \sum_{m_-^{\alpha}=1}^{\infty} \sum_{m_+^{\beta}=1}^{\infty} p_M^{\alpha\beta}(0, m_+^{\beta}) p_M^{\alpha\beta}(m_-^{\alpha}, 0), \quad (5.219)$$

$$S_3 \equiv \sum_{m_+^{\beta}=1}^{\infty} p_M^{\alpha\beta}(0, m_+^{\beta}) p_M^{\alpha\beta}(0,0) = \sum_{m_-^{\beta}=1}^{\infty} p_M^{\alpha\beta}(0,0) p_M^{\alpha\beta}(0, m_-^{\beta}), \quad (5.220)$$

$$S'_3 \equiv \sum_{m_+^{\alpha}=1}^{\infty} p_M^{\alpha\beta}(m_+^{\alpha}, 0) p_M^{\alpha\beta}(0,0) = \sum_{m_-^{\alpha}=1}^{\infty} p_M^{\alpha\beta}(0,0) p_M^{\alpha\beta}(m_-^{\alpha}, 0), \quad (5.221)$$

$$\mathcal{N} \equiv S_0 + 2S_1 + 2S_3 + 2S'_3, \quad (5.222)$$

so that

$$p_{00}^{\alpha\beta} = \frac{S_0}{\mathcal{N}}, \quad (5.223)$$

$$p_{++}^{\alpha\beta} = \frac{S_1}{\mathcal{N}} = p_{--}^{\alpha\beta}, \quad (5.224)$$

$$p_{+-}^{\alpha\beta} = \frac{S_2}{\mathcal{N}} = p_{-+}^{\alpha\beta}, \quad (5.225)$$

$$p_{0+}^{\alpha\beta} = \frac{S_3}{\mathcal{N}} = p_{0-}^{\alpha\beta}, \quad (5.226)$$

$$p_{+0}^{\alpha\beta} = \frac{S'_3}{\mathcal{N}} = p_{-0}^{\alpha\beta}. \quad (5.227)$$

All these sums are computed by inverting the order of the sums over l_{\pm} and m_{\pm} and adding the term corresponding to $m_{\pm} = 0$, for example

$$\sum_{m_+^{\alpha}, m_+^{\beta}=1}^{\infty} \sum_{l_+=\max(m_+^{\alpha}, m_+^{\beta})}^{\infty} \longrightarrow \sum_{l_+=0}^{\infty} \sum_{m_+^{\alpha}, m_+^{\beta}=0}^{l_+} - \text{terms with } m_+ = 0. \quad (5.228)$$

This gives:

$$S_0 = \exp \left\{ -\alpha' k [1 - (1 - q^{\alpha})(1 - q^{\beta})] \right\} \quad (5.229)$$

$$S_1 = \exp \left\{ -\frac{\alpha' k}{2} [1 - (1 - q^{\alpha})(1 - q^{\beta})] \right\} \times \left\{ 1 - \exp \left[-\frac{\alpha' k}{2} q^{\alpha} \right] - \exp \left[-\frac{\alpha' k}{2} q^{\beta} \right] \right\} + \\ + \exp \left\{ -\alpha' k [1 - (1 - q^{\alpha})(1 - q^{\beta})] \right\} \quad (5.230)$$

$$S_2 = \left\{ \exp \left[-\frac{\alpha' k}{2} q^{\alpha} \right] - \exp \left[-\frac{\alpha' k}{2} (1 - (1 - q^{\alpha})(1 - q^{\beta})) \right] \right\} \\ \times \left\{ \exp \left[-\frac{\alpha' k}{2} q^{\beta} \right] - \exp \left[-\frac{\alpha' k}{2} (1 - (1 - q^{\alpha})(1 - q^{\beta})) \right] \right\} \quad (5.231)$$

$$S_3 = \exp \left\{ -\frac{\alpha' k}{2} [1 - (1 - q^{\alpha})(1 - q^{\beta}) + q^{\alpha}] \right\} - \exp \left\{ -\alpha' k [1 - (1 - q^{\alpha})(1 - q^{\beta})] \right\} \quad (5.232)$$

$$S'_3 = \exp \left\{ -\frac{\alpha' k}{2} [1 - (1 - q^{\alpha})(1 - q^{\beta}) + q^{\beta}] \right\} - \exp \left\{ -\alpha' k [1 - (1 - q^{\alpha})(1 - q^{\beta})] \right\} \quad (5.233)$$

$$\begin{aligned} \mathcal{N} = & 2 \exp \left\{ -\frac{\alpha' k}{2} [1 - (1 - q^\alpha)(1 - q^\beta)] \right\} \times \left\{ 1 - \exp \left[-\frac{\alpha' k}{2} q^\alpha \right] - \exp \left[-\frac{\alpha' k}{2} q^\beta \right] \right\} + \\ & + 2 \exp \left\{ -\frac{\alpha' k}{2} (q^\alpha + q^\beta) \right\} + \exp \left\{ -\frac{\alpha' k}{2} [1 - (1 - q^\alpha)(1 - q^\beta)] \right\} \end{aligned} \quad (5.234)$$

The self-consistency equations (5.203) and (5.204) are then

$$q^\alpha = \left\{ \frac{1}{2} \left[1 - \frac{S_0 + 2S_3}{\mathcal{N}} \right] \right\}^{k-1}, \quad (5.235)$$

$$q^\beta = \left\{ \frac{1}{2} \left[1 - \frac{S_0 + 2S'_3}{\mathcal{N}} \right] \right\}^{k-1}. \quad (5.236)$$

Notice that these equations are coupled, as S_0 , S_3 and S'_3 contain both q^α and q^β .

Solution of the self-consistency equations

These equations have four fixed points, of which for $\alpha \rightarrow \infty$ only one is stable. To see it, I consider that as $\alpha \rightarrow \infty$, also $\alpha' \rightarrow \infty$ (I shall verify this later). Then, keeping only the leading exponential term in α' ,

$$S_0 \ll S_3, S'_3, \quad (5.237)$$

$$S_3 \sim \exp \left\{ -\frac{\alpha' k}{2} [1 - (1 - q^\alpha)(1 - q^\beta) + q^\alpha] \right\}, \quad (5.238)$$

$$S'_3 \sim \exp \left\{ -\frac{\alpha' k}{2} [1 - (1 - q^\alpha)(1 - q^\beta) + q^\beta] \right\}, \quad (5.239)$$

$$\mathcal{N} \sim 2 \exp \left\{ -\frac{\alpha' k}{2} [1 - (1 - q^\alpha)(1 - q^\beta)] \right\}. \quad (5.240)$$

The self consistency equations then decouple:

$$q^\alpha = \left\{ \frac{1}{2} \left[1 - \exp \left(-\frac{\alpha' k}{2} q^\alpha \right) \right] \right\}^{k-1}, \quad (5.241)$$

$$q^\beta = \left\{ \frac{1}{2} \left[1 - \exp \left(-\frac{\alpha' k}{2} q^\beta \right) \right] \right\}^{k-1}. \quad (5.242)$$

These equations are identical. Each admits two solutions: one for $q \simeq 0$, and one for $q \simeq 1/2^{k-1}$ (of course, $q = 0$ is also a solution, but a trivial one). The solution close to 0 is

$$q_0 = \left(\frac{\alpha' k}{4} \right)^{-\frac{k-1}{k-2}} + \dots \quad (5.243)$$

and it is unstable, since the derivative of the right hand side is larger than 1. The other solution is

$$q^* = \frac{1}{2^{k-1}} \left\{ 1 - (k-1) \exp \left[-\frac{\alpha' k}{2^k} \right] \right\} + \dots \quad (5.244)$$

and this solution is stable. Therefore, for $\alpha \rightarrow \infty$ we shall have $q^\alpha = q^\beta = q^*$.

The computation of α' as a function of α is similar as the one I've shown in Section 5.3. We must impose that the average total number of occurrences of the new variable be

$$\langle l_+ + l_- \rangle_{\text{Sat}} = \alpha k. \quad (5.245)$$

The distribution of (l_+, l_-) conditioned on the formula being satisfiable is obtained by summing over the values of m_\pm that give no contradictions, i.e.

$$P_{\text{Sat}}(l_+, l_-) = \frac{1}{\mathcal{N}} \left\{ P_M^{\alpha\beta}(0, 0|l_+) P_M^{\alpha\beta}(0, 0|l_-) + \sum_{m_+^\alpha, m_+^\beta=1}^{l_+} P_M^{\alpha\beta}(m_+^\alpha, m_+^\beta|l_+) P_M^{\alpha\beta}(0, 0|l_-) + \right. \\ \left. + \cdots + \sum_{m_-^\alpha=1}^{l_-} P_M^{\alpha\beta}(0, 0|l_+) P_M^{\alpha\beta}(m_-^\alpha, 0|l_-) \right\} \quad (5.246)$$

$$= \frac{1}{\mathcal{N}} \frac{1}{(l_+)!(l_-)!} \left(\frac{\alpha'k}{2} \right)^{l_++l_-} e^{-\alpha'k} \times [(1-q^\alpha)^{l_++l_-} - (1-q^\alpha)^{l_+} - 1 - q^\alpha)^{l_-}] \times \\ \times [(1-q^\beta)^{l_++l_-} - (1-q^\beta)^{l_+} - (1-q^\beta)^{l_-}] \quad (5.247)$$

where the normalization factor \mathcal{N} is the one from (5.234). We obtain:

$$\langle l_+ + l_- \rangle_{\text{Sat}} = \alpha'k \times \frac{1}{\mathcal{N}} \times e^{-\alpha'k} \times \\ \times \left\{ (1-q^\alpha)(1-q^\beta) \exp[\alpha'k(1-q^\alpha)(1-q^\beta)] + \right. \\ + (2-q^\alpha-q^\beta) \exp\left[\frac{\alpha'k}{2}(2-q^\alpha-q^\beta)\right] + \\ + [1 + (1-q^\alpha)(1-q^\beta)] \exp\left[\frac{\alpha'k}{2}[1 + (1-q^\alpha)(1-q^\beta)]\right] + \\ - (1-q^\alpha)(2-q^\beta) \exp\left[\frac{\alpha'k}{2}(1-q^\alpha)(2-q^\beta)\right] + \\ \left. - (2-q^\alpha)(1-q^\beta) \exp\left[\frac{\alpha'k}{2}(2-q^\alpha)(1-q^\beta)\right] \right\}. \quad (5.248)$$

For $\alpha \rightarrow \infty$ we shall have $q^\alpha = q^\beta = q^*$, and the leading order term in the numerator is the one containing $1 + (1-q^*)^2$:

$$\langle l_+ + l_- \rangle_{\text{Sat}} \sim \alpha'k \times \frac{1}{\mathcal{N}} \times [1 + (1-q^*)^2] \exp\left\{-\frac{\alpha'k}{2}[1 - (1-q^*)^2]\right\}, \quad (5.249)$$

with

$$\mathcal{N} \sim 2 \exp\left\{-\frac{\alpha'k}{2}[1 - (1-q^*)^2]\right\} \quad (5.250)$$

so that

$$\langle l_+ + l_- \rangle_{\text{Sat}} = \frac{1}{2} \alpha'k [1 + (1-q^*)^2] + e^{-O(\alpha')} \quad (5.251)$$

and

$$\alpha' = \frac{2\alpha}{1 + (1-q^*)^2} + e^{-O(\alpha)}. \quad (5.252)$$

Uniqueness of the state

The joint probabilities are given, for large α , by

$$p_{00}^{\alpha\beta} = \frac{S_0}{\mathcal{N}} \sim \frac{1}{2} \exp\left\{-\frac{\alpha'k}{2}\left[1 - \left(1 - \frac{1}{2^{k-1}}\right)^2\right]\right\}, \quad (5.253)$$

$$p_{++}^{\alpha\beta} = p_{--}^{\alpha\beta} = \frac{S_1}{\mathcal{N}} \sim \frac{1}{2} - e^{-O(\alpha)}, \quad (5.254)$$

$$p_{0+}^{\alpha\beta} = p_{0-}^{\alpha\beta} = p_{+0}^{\alpha\beta} = p_{-0}^{\alpha\beta} = \frac{S_3}{\mathcal{N}} \sim \frac{1}{2} \exp\left[-\frac{\alpha'k}{2^k}\right]. \quad (5.255)$$

This confirms that the off-diagonal terms are exponentially suppressed, and that $p_{00}^{\alpha\beta} \ll p_{++}^{\alpha\beta}, p_{--}^{\alpha\beta}$. Apart from a fraction of variables of order $e^{-O(\alpha)}$ we see that the variables are constrained and must take the same value in the two states α and β , so that there is actually only one unique state.

The solution to the saddle point equations that we found in Section 5.2 is therefore unique.

5.6 Discussion of the results and conclusion

In Paragraph 5.4.4 I have drawn the conclusion of this work: that the proof of convergence of WP provided in [75] for formulæ extracted from the Planted Ensemble can be extended to formulæ extracted from the Satisfiable Distribution. As we have seen, this contradicts a probabilistic version of Hypothesis 2. There are two questions that remain open and deserve attention.

The first regards Feige's complexity result. Theorem 1 was based on a deterministic form of Hypothesis 2, which is weaker than the probabilistic version refuted by the previous results. It would be very interesting to understand whether the hypotheses of Theorem 1 can be relaxed, and some conclusion reached on the basis of the refutation of Hypothesis 2_p.

Even more interesting, from the physicist's point of view, is the second question. The above discussion for k -SAT can be easily extended to other models, such as k -XORSAT. The characterization of the solutions to large α satisfiable formulæ in terms of the distribution of fields can be repeated, with similar results: that a fraction $1 - e^{-O(\alpha)}$ of the variables are constrained to take a unique value in all the solutions, and that the fields acting on the variables are of order $O(\alpha)$. However, there is a crucial distinction between k -SAT and k -XORSAT: the correlation between the sign of the field acting on a variable and a bias in the number of occurrences between it and its negation, which is present in k -SAT, cannot be present in k -XORSAT for obvious symmetry reasons. Since this is a crucial ingredient of the convergence of WP, it should not be expected to apply to k -XORSAT. It would then be very interesting to find an algorithm which identifies satisfiable k -XORSAT formulæ at large α , and to understand the implications this would have on Theorem 1.

Acknowledgements

I would have never been able to start this work — let alone complete it — without the support and help of many persons, to whom I am deeply indebted and grateful, and whom I wish to thank: Susanna Federici, to whom this work is dedicated; my family and friends, and especially Giulia, Luca and Valentina, for their love and support; Irene and Andrea, for their crucial initial encouragement; Giorgio, Rémi and Francesco, who taught me all I know in this field, and whom I have now the privilege to consider friends; the Laboratoire de Physique Théorique at the Ecole Normale Supérieure in Paris, for its warm welcome, and especially Simona, Nicolas Sourlas who accepted to be my official cotutor, as well as Guilhem and Andrea; and finally, to Silvio Franz, who accepted to referee this thesis.

List of notations

\equiv	Identical to
\sim	Asymptotically equal to, leading order in asymptotic expansions
\simeq	Approximately equal to
$n \div m$	Integer division of n by m
$\mathbb{P}[\cdot]$	Probability
$\mathbb{E}[\cdot]$	Expected value
$\mathbb{I}[\text{event}]$	Indicator function of event , equal to 1 if event is true and 0 otherwise
\vee	Logical OR
\wedge	Logical AND
\oplus	Logical XOR
$ \mathcal{S} $	Cardinality of set \mathcal{S}
$\langle \cdot \rangle$	Thermodynamic average
\overline{O}	Average over disorder of O
i, j, k, \dots	Site indices from 1 to N
a, b, c, \dots	Replica indices from 1 to n
σ_i	Individual spin
σ	N -component spin configuration
$\boldsymbol{\sigma}$	Replicated $N \times n$ spin configuration
σ^a	N -component spin configuration of replica a
$\vec{\sigma}_i$	n -component spin configuration on site i
$\vec{\sigma}, \vec{\tau}$	Generic n -component spin configurations
σ_i^a	Value of spin on site i for replica a
α	Ratio between number of clauses M and number of variables N in a boolean constraint satisfaction problems
α_s	Threshold value for SAT/UNSAT transition
α_c	Threshold value for clustering transition
α_0	Lower bound on α_s from the second moment inequality
α_h	Largest value of α for which a poissonian DPLL heuristic succeeds with positive probability
Σ_c	Clustering transition surface
Σ_s	SAT/UNSAT transition surface
Σ_k	Critical surface (i.e. intersection of Σ_c and Σ_s)
Σ_q	Contradiction surface
\mathcal{F}	k -SAT formula
$\mathcal{P}_{\text{Unif}}[\mathcal{F}]$	Uniform measure over random formulæ
$\mathcal{P}_{\text{Sat}}[\mathcal{F}]$	Uniform measure over satisfiable formulæ
$\mathcal{P}_{\text{Unif}}[\mathcal{F}]$	Planted measure over random formulæ

$c(\vec{\sigma})$	Fraction of sites with replicated configuration $\vec{\sigma}$, functional order parameter
$R(h)$	Distribution of fields, functional order parameter equivalent to $c(\vec{\sigma})$
\mathcal{F}	Free energy density functional
ν	“Thermodynamic potential”, $\nu \equiv \beta n$ as $\beta \rightarrow \infty$ and $n \rightarrow 0$
$\epsilon_0(\nu)$	Ground state energy density of formulæ conditioned on ν
r_p	Weight of $R(h)$ over $h = p \in \mathbb{Z}$
$I_p(x)$	Modified Bessel function of integer order
ρ_p	Limit of r_p for $\nu \rightarrow \infty$

Bibliography

- [1] M. Mézard, G. Parisi and M.A. Virasoro, *Spin Glass Theory and Beyond*, Lecture notes in Physics (Vol. 9), World Scientific (1987)
- [2] C.H. Papadimitriou, *Computational Complexity*, Addison-Wesley (1998)
- [3] G. Parisi, *Statistical Field Theory*, Frontiers in Physics (Vol. 66), Perseus (1988)
- [4] S.F. Edwards and P.W. Anderson, *J. Phys.* **F 5** 965 (1975)
- [5] D. Sherrington and S. Kirkpatrick, *Phys. Rev. Lett.* **35** 1792 (1975)
- [6] D.J. Thouless, P.W. Anderson, R.G. Palmer, *Phil. Mag.* **35 3** 593 (1977)
- [7] V. Cannella and J.A. Mydosh, *Phys. Rev.* **B 6** 4220 (1979)
- [8] J.L. Tholence and R. Tournier, *Journal de Physique (Paris)* **35 C** 4 (1974)
- [9] S. Nagato, P.H. Keeson and H.R. Harrison, *Phys. Rev.* **B 19** 1663 (1979)
- [10] L.-M. Martinez and C.A. Angell, *Nature* **410** 663 (2001)
- [11] R. Richert and C.A. Angell, *J. Chem. Phys.* **108** 9016 (1998)
- [12] A. Crisanti and H.-J. Sommers, *Z. Phys.* **B 87** 391 (1992)
- [13] A. Crisanti, H. Horner, H.-J. Sommers, *Z. Phys.* **B 92** 257 (1993)
- [14] A. Barrat, (Unpublished)
Available on: [arXiv:cond-mat/9701031](https://arxiv.org/abs/cond-mat/9701031)
- [15] G. Parisi, *Phys. Rev. Lett.* **43** 1754 (1979)
- [16] G. Parisi, *J. Phys.* **A 13** 1101 (1980)
- [17] G. Parisi, *Phys. Rev. Lett.* **50** 1946 (1983)
- [18] L. Viana and A.J. Bray, *J. Phys.* **C 18** 3037 (1985)
- [19] M. Mézard and G. Parisi, *Journal de Physique - Lettres* **46 17** 771 (1985)
- [20] H. Orland, *Journal de Physique - Lettres* **46 17** 763 (1985)
- [21] C. De Dominicis and P. Mottishaw, *J. Phys.* **A 20** 1267 (1987)
- [22] R. Monasson, *J. Phys.* **A 31** 513 (1998)

- [23] A. Turing, *Proc. London Math. Soc.* **2** **42** 230 (1936)
- [24] S. Cook, *Proceedings of the 3rd annual ACM Symposium on the Theory of Computing* 151 (1971)
- [25] L. Levin, *Problemy Peredachi Informatsii* **9** (**3**) 265 (1973)
- [26] S. Arora, Ph.D. Dissertation
Available on: <http://www.cs.princeton.edu/arora/pubs/thesis.pdf>
- [27] S. Cocco, R. Monasson, A. Montanari and G. Semerjian, in *Computational Complexity and Statistical Physics*, edited by G. Istrate, C. Moore and A. Percus, Oxford University Press (2006)
- [28] C.H. Papadimitriou, *Proceedings of the 32nd annual IEEE symposium on the foundations of computer science*, 163 (1991)
- [29] U. Schöningh, *Algorithmica* **32** 615 (2002)
- [30] M. Alekhnovich and E. Ben-Sasson, *Proceedings of the 44th Annual Symposium on Foundations of Computer Science* (2003)
- [31] G. Semerjian and R. Monasson, *Phys. Rev.* **E** **67** 066103 (2003)
- [32] W. Barthel, A.K. Hartmann and M. Weigt, *Phys. Rev.* **E** **67** 066104 (2003)
- [33] E. Aurell, U. Gordon and S. Kirkpatrick, *Eighteenth Annual Conference on Neural Information Processing Systems* (2004)
- [34] A. Montanari, G. Parisi and F. Ricci-Tersenghi, *J. Phys.* **A** **37** 2072 (2004)
- [35] M. Davis and H. Putnam, *J. of the ACM* **7** (**1**) 201 (1960)
- [36] M. Davis, G. Logemann and D. Loveland, *Comm. of the ACM* **5** (**7**) 394 (1962)
- [37] S. Cocco and R. Monasson, *Eur. Phys. J.* **B** **22** 505 (2001)
- [38] S. Cocco and R. Monasson, *Theor. Comp. Sci.* **320** 345 (2004)
- [39] M.T. Chao and J. Franco, *SIAM J. of Computing* **15** 1106 (1986)
- [40] M.T. Chao and J. Franco, *Information Science* **51** 289 (1990)
- [41] A. Frieze and S. Suen, *J. of Algorithms* **20** 312 (1996)
- [42] D. Achlioptas, *Theor. Comp. Sci.* **265** 159 (2001)
- [43] N.C. Wormald, *Ann. Appl. Prob.* **5** (**4**) 1217 (1995)
- [44] J. Pearl, *Proceedings of the American Association of Artificial Intelligence National Conference on AI* (1982)
- [45] M. Mézard and G. Parisi, *Eur. Phys. J.* **B** **20** 217 (2001)
- [46] M. Mézard and G. Parisi, *J. Stat. Phys.* **111** 1 (2002)
- [47] M. Mézard, G. Parisi and R. Zecchina, *Science* **297** 812 (2002)

- [48] M. Mézard and R. Zecchina, *Phys. Rev. E* **66** 056126 (2002)
- [49] P. Erdős and A. Rényi, *Publ. Math. Debrecen* **6** 290 (1959)
- [50] P. Erdős and A. Rényi, *Publ. Math. Inst. Hungar. Acad. Sci.* **5** 17 (1960)
- [51] S. Kirkpatrick and R. Swendsen, *Comm. ACM* **28-4** 363 (1985)
- [52] S. Kirkpatrick and B. Selman, *Science* **264** 1297 (1994)
- [53] D. Mitchell, B. Selman and H. Levesque, *Proceedings of the 10th National Conference on Artificial Intelligence* 459 (1992)
- [54] B. Selman and S. Kirkpatrick, *Art. Intell.* **81** 273 (1996)
- [55] N. Creignou and H. Daudé, *Discr. Appl. Math.* **96-97** 41 (1999)
- [56] S. Cocco, O. Dubois, J. Mandler and R. Monasson, *Phys. Rev. Lett.* **90** 047205 (2003)
- [57] M. Mézard, F. Ricci-Tersenghi and R. Zecchina, *J. Stat. Phys.* **111** 505 (2003)
- [58] T. Mora and M. Mézard, *J. Stat. Mech.* P10007 (2006)
- [59] E. Friedgut, *J. Amer. Math. Soc.* **12-4** 1017 (1999)
- [60] D. Achlioptas, A. Naor and Y. Peres, *Nature* **435** 759 (2005)
- [61] S. Mertens, M. Mézard and R. Zecchina, *Rand. Struct. Algo.* **28** 340 (2006)
- [62] M. Mézard, T. Mora and R. Zecchina, *Phys. Rev. Lett.* **94** 197205 (2005)
- [63] H. Daudé, M. Mézard, T. Mora and R. Zecchina, Submitted to *Theor. Comp. Sci.*
Preprint: [arXiv:cond-mat/0506053v3](https://arxiv.org/abs/cond-mat/0506053v3)
- [64] G. Biroli, R. Monasson and M. Weigt, *Eur. Phys. J. B* **14** 551 (2000)
- [65] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian and L. Zdeborova, *PNAS* **107** 10318 (2007)
- [66] G. Semerjian, Accepted by *J. Stat. Phys.*
Preprint: [arXiv/0705.2147](https://arxiv.org/abs/0705.2147)
- [67] F. Altarelli, R. Monasson and F. Zamponi, *Proceedings of the International Workshop on Statistical Mechanics Informatics*, Kyoto 2008 (to be published)
Preprint: [arXiv:0709.0367v1](https://arxiv.org/abs/0709.0367v1) [Cs.CC]
- [68] G. Parisi, *Proceedings of the Oskar Klein Centennial Symposium*, World Scientific, 60 (1995)
- [69] S. Franz and G. Parisi, *Journal de Physique* **I 5** 1401 (1995)
- [70] S. Franz and G. Parisi, *Phys. Rev. Lett.* **79** 2486 (1997)
- [71] S. Franz and G. Parisi, *Phil. Mag.* **B 77** 239 (1998)
- [72] S. Franz and G. Parisi, *Physica A* **261** 317 (1998)
- [73] R. Monasson, *Phys. Rev. Lett.* **75** 2847 (1995)

- [74] U. Feige, *Proceedings of the 4th STOC meeting* 534 (2002)
Available on: <http://www.wisdom.weizmann.ac.il/~feige/approx.html>
- [75] U. Feige, E. Mossel and D. Vilenchik, *Proceedings of the Random 2006 Conference* 339 (2006)
Available on: http://research.microsoft.com/research/theory/feige/homepagefiles/WP_9_14.ps
- [76] F. Altarelli, R. Monasson and F. Zamponi, *J. Phys.* **A** **40** 867 (2007)
Available on: [arXiv:cs/0609101v2](http://arxiv.org/abs/cs/0609101v2) [Cs.CC]
- [77] R. Monasson and R. Zecchina, *Phys. Rev.* **E** **56** 1357 (1997)